2019

A Lightweight Privacy-Preserving Fair Meeting Location Determination Scheme

Hua Shen
University of Wollongong, huas@uow.edu.au

Mingwu Zhang
csmwzhang@gmail.com

Hao Wang
University of Wollongong, haow@uow.edu.au

Fuchun Guo
University of Wollongong, fuchun@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

A Lightweight Privacy-Preserving Fair Meeting Location Determination Scheme

Abstract

Equipped with mobile devices, people relied on location-based services can expediently and reasonably organize their activities. But location information may disclose people's sensitive information, such as interests, health status. Besides, the limited resources of mobile devices restrict the further development of location-based services. In this paper, aiming at the fair meeting position determination service, we design a lightweight privacy-preserving solution. In our scheme, mobile users only need to submit service requests. A cloud server and a location services provider are responsible for service response, where the cloud server achieves most of the calculation, and the location services provider determines the fair meeting location based on the computational results of the cloud server and broadcasts it to mobile users. The proposed scheme adopts homomorphic encryptions and random permutation methods to preserve the location privacy of mobile users. The security analyses show that the proposed scheme is privacy-preserving under our defined threat models. Besides, the presented solution only needs to calculate n Euclidean distances, and hence, our scheme has linear computation and communication complexity.

Keywords

privacy-preserving, fair, determination, location, lightweight, scheme, meeting

Disciplines

Engineering | Science and Technology Studies

Publication Details

Shen, H., Zhang, M., Wang, H., Guo, F. & Susilo, W. (2019). A Lightweight Privacy-Preserving Fair Meeting Location Determination Scheme. IEEE Internet of Things Journal, 14 (8), 1-11.

A Lightweight Privacy-Preserving Fair Meeting Location Determination Scheme

Hua Shen, Mingwu Zhang, Hao Wang, Fuchun Guo, and Willy Susilo *Senior Member, IEEE*

***Abstract*—Equipped with mobile devices, people relied on location-based services can expediently and reasonably organize their activities. But location information may disclose people's sensitive information, such as interests, health status. Besides, the limited resources of mobile devices restrict the further development of location-based services. In this paper, aiming at the fair meeting position determination service, we design a lightweight privacy-preserving solution. In our scheme, mobile users only need to submit service requests. A cloud server and a location services provider are responsible for service response, where the cloud server achieves most of the calculation, and the location services provider determines the fair meeting location based on the computational results of the cloud server and broadcasts it to mobile users. The proposed scheme adopts homomorphic encryptions and random permutation methods to preserve the location privacy of mobile users. The security analyses show that the proposed scheme is privacy-preserving under our defined threat models. Besides, the presented solution only needs to calculate n Euclidean distances, and hence, our scheme has linear computation and communication complexity.**

***Index Terms*—Privacy-preserving, location privacy, location-based service, fair meeting location.**

I. INTRODUCTION

NOWADAYS, the rapid development of location-aware technologies such as mobile communication and sensing devices and the widespread use of smart mobile devices (e.g., smartphones) can obtain accurate location information of users at any time. Location-based services (LBSs) become very popular in almost all social, business, and industrial domains [1]–[3]. Users need to send their location information and queries to the LBS server, and then they can enjoy the corresponding services provided by the LBS server. Some typical LBSs include "check-in," location sharing, near friends' query, map information, automotive traffic monitoring, and road navigation [4], [5]. For example, users can find the nearest restaurant, petrol station, market, hospital, and cinema through "check-in" services [6]. In the industrial sector, through "check-in" services, factories can find the nearest partner who provides a similar manufacturing process. The "check-in" services do not depend on the locations of other users [7]. The users in a group can obtain a fair meeting location for the whole group through location sharing services [8], which rely on the sharing of places (or location preferences) by group users [7]. Users can download various LBS applications from application stores such as the Apple Store or Google Play Sore or Huawei AppGallery [9]. LBSs bring convenience to users, while they also bring substantial economic benefits. According to a survey by Pyramid Research [10], location-based services had a 10.3 billion dollars market in 2015. But LBSs raise the danger of revealing private personal information (such as the home address, the current location, the history locations, and so on). The analysis of the location information poses threats to reveal sensitive information of users (such as economic status, living habits, health conditions, social relationships, etc.). For instance, in the Uber application, users need to share their current and target locations to the Uber service provider. If the service provider is curious, it could easily infer users' home address, travel habits (such as what time to go out, what time to go home, car model selection preference), health conditions (for example, whether recently is or not ill), and so on. User study in [7] shows nearly 88% of users are not comfortable with sharing their location information. Therefore, how to protect users' privacy in location sharing services is a crucial issue.

Fair Meeting Location Determination problem is a specific problem in LBSs. The problem is to determine a location from a given set of user locations (or location preferences) as the meeting place such that it is fair to all users in the group. "Fair" in the problem means that the determined meeting location cannot be too far away from nor close to some users. The privacy issue in this problem is representative of the relevant privacy threats in LBSs [7]. In this paper, considering the constrained resource of mobile devices (e.g., smartphones), we focus on how to high-efficiently resolve the Fair Meeting Location Determination problem in a privacy-preserving manner. Our goal is to enable each user in a group to obtain a fair meeting location with low computation and communication costs, and without disclosing any user's

Manuscript received June 23, 2019. This work is supported by the National Natural Science Foundation of China (61702168, 61672010), the Hubei Provincial Department of Education Key Project (D20181402), the open research project of The Hubei Key Laboratory of Intelligent Geo-Information Processing (KLIGIP-2017A11).

Corresponding authors: Hua Shen and Mingwu Zhang.

H. Shen and M. Zhang are with the School of Computers, Hubei University of Technology, Wuhan, Hubei 430068, China (e-mail: cshshen@hbut.edu.cn, csmwzhang@gmail.com).

H. Wang is with the school of Information Science and Engineering, Shandong Normal University, Jinan, Shandong 250014, China (e-mail: wanghao@sdnu.edu.cn)

F. Guo and W. Susilo are with the School of Computing and Information Technology, Institute of Cybersecurity and Cryptology, University of Wollongong, NSW 2522, Australia (e-mail: {fuchun, wsusilo}@uow.edu.au).

H. Shen and H. Wang are also with the School of Computing and Information Technology, Institute of Cybersecurity and Cryptology, University of Wollongong, NSW 2522, Australia.

M. Zhang is also with the State Key Laboratory of Cryptology, P. O. Box 5159, Beijing 100878, China, and with the School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin, Guangxi 541004, China.

TABLE I
COMPARISON OF THREE CATEGORIES SCHEMES

	Data Distortion	TTP	Comm. Overhead	Comp. Cost	Protection Strength
Obfuscation	$\surd$	$\times$	Low	Middle	Middle
Anonymity	$\times$	$\surd$	High	Middle	Low
Encryption	$\times$	$\times$	Middle	High	High

location information. For achieving the goal, we propose a lightweight privacy-preserving fair meeting location determination scheme (FMLD).

The remainder of the paper is organized as follows. We overview related work in Section II. In Section III, we formulate the problem of the fair meeting location determination and describe the system model, the threat model, and the design goal, finally summarize our contributions. In Section IV, we recall Paillier encryption. Then, we show the details of the proposed scheme in Section V, followed by its correctness analysis, security proof, and performance analysis in Sections VI, VII and VIII, respectively. Finally, we summarize our work in Section IX.

II. RELATED WORK

Recently, many approaches have been proposed to tackle the privacy threats in LBS applications. We roughly divide them into three categories: obfuscation based solutions, anonymity based solutions, and encryption based resolutions [11]–[14]. For preventing revealing the exact location to others, obfuscation based solutions usually add noise to places or quantize locations [11], but which reduces the quality of the location data and LBSs [11], [15]. Anonymity based schemes address privacy concerns of users through separating users' identities from their locations by using the pseudo-ID technique [16], k-anonymity [11], [17], [18], dummy locations [9], [19], [20], and others. But anonymity based schemes bring high communication cost [13] and usually rely on a trusted third party (namely anonymizer), which may suffer from a single point of attack [11], [21]. In encryption based schemes [7], [8], [13], [22], a user uses homomorphic encryption to encrypt his/her location information and sends the encrypted data to a cloud server, or an LBS server. Then the server returns the result to the user. Encryption based schemes are with low communication costs, but their computation costs are too high to be suitable for smartphones. The overview of the comparison between the three categories schemes is illustrated in Table I.

Because the Fair Meeting Location Determination problem requires to provide the accurate meeting location to users and avoiding the emergence of a trust third party, we pay attention to how to leverage homomorphic encryption to resolve the Fair Meeting Location Determination problem efficiently. In 2014, Bilogrevic et al. [7] proposed a privacy-preserving two-party computation framework to resolve the problem and then presented two concrete schemes to realize the frame. The one scheme was realized through employing BGN cryptosystem and ElGamal cryptosystem, and the other was achieved by utilizing Paillier cryptosystem and ElGamal cryptosystem. However, there are some deficiencies in Bilogrevic's schemes. First, the schemes are not practical due to the heavy computation burden of user-side and the limited resources of mobile devices. Take the scheme realized by using Paillier and ElGamal cryptosystems as example, each user need to carry out $n+1$ Paillier encryption operations, n Paillier decryption operations, 2 ElGamal encryption operations, and $n-1$ ElGamal decryption operations, where n is the number of users. Second, there are $n(n-1)/2$ Euclidean distances needed to be calculated over ciphertext space, which will lay a heavy burden on the mobile devices. To reduce the computational overheads in users' resource-constrained mobile devices, Wang [8] introduced an untrusted cloud server to handle most computing tasks in their scheme. Wang's scheme takes advantage of the homomorphic property of BGN and ElGamal cryptosystems to determinate a fair meeting location from a set of users' current or preferred positions without revealing their privacy. However, Wang's scheme similarly needs to compute $n(n-1)/2$ Euclidean distances over ciphertext space, which results in the scheme being inefficient. Besides, in Wang's scheme, the LDS server (named MLDS) finally broadcasts the series number of the chosen location as the fair meeting location rather than the location information, so users still do not know where is the meeting location selected. In addition, MLDS obtains the Euclidean distance of each pair of users and the Euclidean distance average and variance of each user. MLDS can inference some sensitive information from these data, for example, it can learn whether the locations of the group users are relatively centralized or scattered, and which two users in the group are the furthest apart.

Based on the above analysis, to achieve spending lower computation and communication costs to resolve the Fair Meeting Location Determination problem by using homomorphic encryption, we identify the core issue is how to reduce the number of Euclidean distance calculations. Based on this idea, we propose a lightweight privacy-preserving fair meeting location determination scheme (FMLD).

III. PROBLEM FORMULATION, MODELS AND DESIGN GOAL

In this section, we formulate the fair meeting location determination problem and describe the system and threat models considered in our work, and identify the design goal, finally summarize our main contributions.

A. Problem Formulation

The appropriate meeting location can be determined in several ways, for example, the approach adopted by [7] is to find the location which has the minimum of the maximum Euclidean distance within all locations, the method exploited by [8] is to determine which place has minimum Euclidean distance variance. Fig.1 shows our method to determine the fair meeting location. There are five users in Fig.1, the intersection point of two dashed lines is the geometric center (x, y) of the five users' preferred locations (that is, $(x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4), (x_5, y_5)$), the solid lines represent the Euclidean distances between users' locations

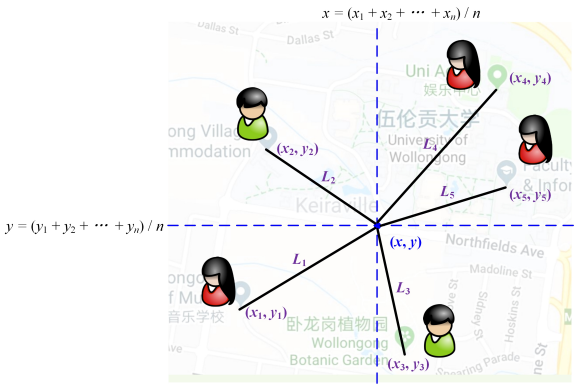

Fig. 1. Problem description.

Algorithm 1 Fire Meeting Location (AL_{FMLD}).

Input: (x_1, y_1), (x_2, y_2), $\cdots$, (x_n, y_n).

Output: the fair meeting location which is chosen from the input locations.

1: Calculate the geometric center (x, y) of n preferred locations: $x = (x_1 + \cdots + x_n)/n$, $y = (y_1 + \cdots + y_n)/n$;

2: Compute the square of the Euclidean distance L_i between $User_i$ preferred meeting location and the geometric center (for all $i \in \{1, \cdots, n\}$): $L_i = (x_i - x)^2 + (y_i - y)^2$

3: Determine the minimum of $\{L_1, L_2, \cdots, L_n\}$, assume it is L_k ($k \in \{1, 2, \cdots, n\}$);

4: **return** (x_k, y_k);

and the geometric center. We use L_i denote the square of the Euclidean distance between ith user's location and the geometric center. Suppose $L_5 = min\{L_1, L_2, L_3, L_4, L_5\}$, the fair rendezvous location is (x_5, y_5). Generalize the above problem description as follows. Assume there are n users, and the preferred meeting position of each user U_i (for $i \in \{1, 2, \cdots, n\}$) is (x_i, y_i).

The main work of this paper is to obtain the output of the above algorithm without revealing its inputs. In this paper, we view (x_i, y_i) as the location privacy of User U_i. Clearly, in our work, we only need to calculate n Euclidean distances securely, but in [7] and [8], the number is $n(n-1)/2$.

The following issue is how to utilize homomorphic encryption to achieve the AL_{FMLD} algorithm (Alg.1). We use E_j and D_j to denote the encryption algorithm and decryption algorithm of a homomorphic cryptosystem under different public and private keys, where the value of j is successively $1, 2, \cdots$ as need. And we denote the encryption algorithm of a broadcast cryptograph as E. The $PPAL_{FMLD}$ algorithm (Alg.2) provides a kind of secure realization of the AL_{FMLD} algorithm (Alg.1).

The inputs of the $PPAL_{FMLD}$ algorithm (Alg.2) are the request for determining a fair meeting location, which is provided by users. The $PPAL_{FMLD}$ algorithm (Alg.2) is the response process, and the outputs of it are the response. Since there are three encryption and decryption algorithms pairs (i.e., (E_1, D_1), (E_2, D_2), (E_3, D_3)) in the $PPAL_{FMLD}$ algorithm (Alg.2), we need three entities to realize the $PPAL_{FMLD}$

Algorithm 2 Private-Preserving FMLD ($PPAL_{FMLD}$).

Input: $(E_1(x_1, y_1)), \cdots, (E_1(x_n, y_n))$; $E_2(x_1), \cdots, E_2(x_n)$; $E_2(y_1), \cdots, E_2(y_n)$

Output: the broadcast ciphertext of the fair meeting location.

1: **for** $i = 1$ to n **do**
2: $E_1(sx_i, sy_i) \leftarrow E_1(x_i, y_i)$; // s is a random noise
3: $E_2(sx_i) \leftarrow E_2(x_i)$;
4: $E_2(sy_i) \leftarrow E_2(y_i)$;
5: **end for**
6: $\{E_1(sx_{p_1^{(1)}}, sy_{p_1^{(1)}}), \cdots, E_1(sx_{p_n^{(1)}}, sy_{p_n^{(1)}})\}$ $\leftarrow_{randomlypermute} \{E_1(sx_1, sy_1), \cdots, E_1(sx_n, sy_n)\}$
7: $\{E_2(sx_{p_1^{(2)}}), \cdots, E_2(sx_{p_n^{(2)}})\}$ $\leftarrow_{randomlypermute} \{E_2(sx_1), \cdots, E_2(sx_n)\}$
8: $\{E_2(sy_{p_1^{(3)}}), \cdots, E_2(sy_{p_n^{(3)}})\}$ $\leftarrow_{randomlypermute} \{E_2(sy_1), \cdots, E_2(sy_n)\}$
9: **for** $i = 1$ to n **do**
10: $sx_{p_i^{(2)}} = D_2(E_2(sx_{p_i^{(2)}}))$;
11: $sy_{p_i^{(3)}} = D_2(E_2(sy_{p_i^{(3)}}))$;
12: **end for**
13: $x' = \sum_{i=1}^{n} sx_{p_i^{(2)}}/n$, $y' = \sum_{i=1}^{n} sy_{p_i^{(3)}}/n$;
14: **for** $i = 1$ to n **do**
15: $E_1(L_i) = Compu_Euc(E_1(sx_{p_i^{(1)}}, sy_{p_i^{(1)}}), E_1(x', y'), x', y')$;
16: **end for**
17: **for** $i = 1$ to n **do**
18: $L_i = D_1(E_1(L_i))$;
19: **end for**
20: $L_k = min\{L_1, \cdots, L_n\}$;
21: $E_3(k)$;
22: $k = D_3(E_3(k))$ and find $E_1(x_{p_k^{(1)}}, y_{p_k^{(1)}})$;
23: $E(D_1(E_1(x_{p_k^{(1)}}, y_{p_k^{(1)}})))$;
24: **return** $E(x_{p_k^{(1)}}, y_{p_k^{(1)}})$;

algorithm (Alg.2) synergistically. We introduce a fog device to achieve steps 1 to 8 and 22, and a cloud server to fulfill steps 9 to 16, a service provider to implement steps 17 to 21 and 23, 24. Our system model is described in the following section detailedly.

B. System Model

As shown in Fig.2, our system model comprises four types of entities: (i) a set of users with mobile devices $\{U_1, U_2, \cdots, U_n\}$, (ii) a fog device, denoted as FD, (iii) a cloud server, indicated as CS, and (iv) a third-party location determination service provider, signified as LDSP. Users provide the inputs of the algorithm mentioned above; FD, CS, and LDSP achieve the algorithm without knowing the inputs; and then LDSP return the output of the algorithm to users, but LDSP doesn't know to whom the location belongs.

Each user can use his current location as his preferred meeting location, or he can specify another location as his preferred location. The position related to U_i is (x_i, y_i), of which the values of coordinates are latitude and longitude. Users can obtain accurate values of positions by using GPS. After executing encryption operation, users send the encrypted

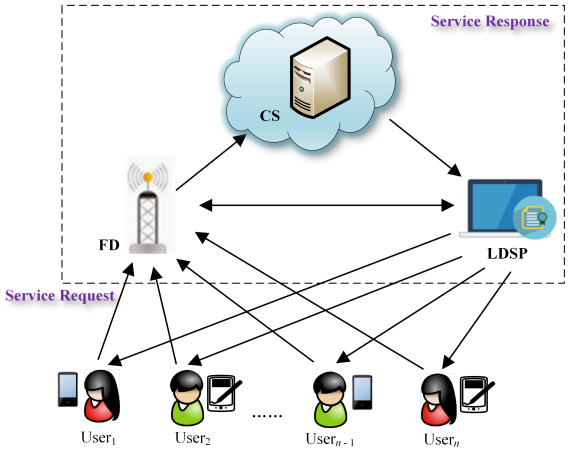

Fig. 2. System model for fair meeting location determination.

preferred locations to FD. FD disturbs the received ciphertexts and then forwards them to CS. After that, CS calculates the geometric center of the candidate locations and the square of Euclidean distances between candidate meeting locations and the geometric center in a privacy-preserving way and then transmits n encrypted Euclidean distances to LDSP. LDSP determines the fair meeting location relied on the received Euclidean distances and broadcasts the result to users.

C. Threat Model

In this paper, our primary goal is to protect the location privacy of users while providing location service for them. The following threat model is assumed:

- FD, CS, and LDSP are honest-but-curious, which honestly follow the underlying scheme, but are curious about the location privacy of mobile users. They try to learn information about the position privacy of the users from the received inputs, the intermediate results, and the computed outputs. Users are also honest-but-curious, who keep the system running smoothly and perhaps try to learn the location preferences of other users from the intermediate results and the response from LDSP. We refer to such attacks as internal attackers' passive attacks. Furthermore, CS and LDSP can collude in an attempt to obtain users' location privacy. But CS or LDSP would not cooperate with FD, and this is because if CS or LDSP agree to collude with FD to obtain the information of a user's position, then FD would obtain some privacy information of CS or LDSP, such as their private keys. Users may cooperate between users for gaining the location information of other users. We call such attacks as internal attackers' active attacks.
- External adversaries can eavesdrop the communication to obtain the transmit reports and could intrude in the databases of FD, CS, and LDSP.

Note that, since the users' location privacy preservation is our focus, some active attacks are beyond the scope of this work.

D. Design Goal

Under the above system model, our design goal is to develop a lightweight privacy-preserving solution for determining a fair meeting location for users using mobile devices. Specifically, we should achieve the following three objectives:

- For users, their location information is sensitive and should be protected. Hence, the proposed scheme should guarantee against the attacking by internal or external adversaries described in the above threat model.
- Although protecting users' location privacy is one of our goals, it should not reduce the service quality of a fair meeting location determination service. Therefore, it is necessary for our presented scheme to offer sound fair meeting location determination service.
- Due to the limited resource of mobile devices, the proposed scheme should not consume many resources of mobile users. Besides, although a cloud server and a providing location-based services server have generous storage and computing resources, they will handle thousands of service requests at the same time. Hence, computational costs and communication burden of server-side should also be as less as possible.

E. Our Contribution

In summary, we make the following contributions:

- We present a novel method to determine a fair meeting location in n candidate meeting locations. The main idea is that firstly we find out the geometrical center of the n meeting locations, and then calculate the Euclidean distances between each candidate meeting location and the geometric center respectively. We take the location with the smallest Euclidean distance as the fair meeting location. Our method only needs to compute n Euclidean distances. Therefore the presented method can reduce computation and communication costs expressively.
- We propose a privacy-preserving scheme, which leverages homomorphic encryption and random permutation skill to achieve the above method. In other words, the proposed scheme can, in a private-preserving manner, obtain the geometric center of n candidate meeting locations and calculate the Euclidean distances between the geometric center and each candidate meeting location.
- The proposed scheme is suitable for mobile user-centric application service. The proposed scheme has linear computation and communication complexity, and a cloud server achieves its main computation tasks. The emerging fog architecture can guarantee excellent user experience [23]. Therefore we adopt a fog device in our system to improve the quality of user experience.

IV. Preliminaries

In this section, we give an overview for Paillier homomorphic encryption [24] which server as the basis of the proposed scheme. The Paillier homomorphic cryptosystem mainly consists of three algorithms:

Key Generation(κ): Given a security parameter κ, choose two κ-bit prime numbers p and q. Let $N = pq$, $\lambda = lcm(p -$

$1, q-1)$, define a function $L(\mu) = (\mu-1)/N$, pick a random generator $g \in \mathbb{Z}^*_{N^2}$, and calculate $\mu = (L(g^\lambda \ mod \ N^2)) - 1$. Set the public key as $PK = (N, g)$ and the private key as $SK = (\lambda, \mu)$.

Encryption(m, PK): Pick a random number $r \in \mathbb{Z}^*_N$ and encrypt a message $m \in \mathbb{Z}_N$ with the public key PK: $C = g^m \cdot r^N \bmod N^2$.

Decryption(C, SK): Consider the ciphertext $C = g^m \cdot r^N \bmod N^2$, and recover the corresponding message with the private key SK: $m = L(C^\lambda \bmod N^2) \cdot \mu$.

Paillier cryptosystem has the following homomorphic properties: $E(m_1, r_1)E(m_2, r_2) = E(m_1 + m_2, r_1 \cdot r_2)$ and $E(m_1, r_1)^{m_2} = E(m_1 \cdot m_2, r_1^{m_2})$.

V. THE PROPOSED SCHEME

This section presents a lightweight privacy-preserving fair meeting location determination scheme (FMLD). Fig.3 illustrates the prime process procedure of FMLD. In Fig.3, we use E_1, E_2, and E_3 denote Paillier encryption based on public keys PK_{SP}, PK_{CS} and PK_{FD}, respectively, and respectively utilize D_1, D_2, and D_3 denote Paillier decryption based on private keys SK_{SP}, SK_{CS} and SK_{FD}. We divide the prime process into three parts: system initialization, service request, and service response; the full description is as follows.

A. System Initialization

In this part, the participants' public and private key pairs and the system public parameters are generated. For each user U_i, $\forall i \in \{1, 2, \cdots, n\}$, the system generates his public and private key pair (pk_i, sk_i) for broadcast encryption, and generates the LDSP's public and private key pair (PK_{SP}, SK_{SP}), the CS's public and private key pair (PK_{CS}, SK_{CS}), and the FD's public and private key pair (PK_{FD}, SK_{FD}) for Paillier encryption respectively. Assume the position coordinates have a value range of T, $T << N$ and the Euclidean distance of arbitrary two positions is smaller than N.

B. Service Request

In this part, users provide their preferred locations to request LDSP return a fair meeting position without jeopardizing their location privacy. The detailed processing is as the following steps:

Step 1. Each user U_i, $\forall i \in \{1, 2, \cdots, n\}$, randomly chooses r_{i1}, r_{i2}, r_{i3}, r_{i4}, r_{i5}, $r_{i6} \in \mathbb{Z}^*_N$ and encrypts the coordinates of his preferred position (x_i, y_i) with the LDSP's public key PK_{SP} as follows:

$$C_{i1} = g^{{x_i}^2} \cdot r_{i1}^N \bmod N^2 \quad (1)$$

$$C_{i2} = g^{N-x_i} \cdot r_{i2}^N \bmod N^2 \quad (2)$$

$$C_{i3} = g^{{y_i}^2} \cdot r_{i3}^N \bmod N^2 \quad (3)$$

$$C_{i4} = g^{N-y_i} \cdot r_{i4}^N \bmod N^2 \quad (4)$$

$$C_{i5} = g^{x_i} \cdot r_{i5}^N \bmod N^2 \quad (5)$$

$$C_{i6} = g^{y_i} \cdot r_{i6}^N \bmod N^2 \quad (6)$$

U_i randomly chooses $r'_{i1}, r'_{i2} \in \mathbb{Z}^*_N$ and encrypts the coordinates of his preferred position (x_i, y_i) with the CS's public key PK_{CS} as follows:

$$C'_{i1} = g^{x_i} \cdot {r'_{i1}}^N \bmod N^2 \quad (7)$$

$$C'_{i2} = g^{y_i} \cdot {r'_{i2}}^N \bmod N^2 \quad (8)$$

Step 2. U_i sends $C_i||C_{i5}||C_{i6}||C'_{i1}||C'_{i2}$ to FD, where $C_i = C_{i1}||C_{i2}||C_{i3}||C_{i4}$.

Note that C_{i1}, C_{i2}, C_{i3}, and C_{i4} are utilized to compute the Euclidean distance, C'_{i1} and C'_{i2} are used to calculate the geometric center, C_{i5} and C_{i6} be leveraged to search the corresponding ciphertext of the location determined.

C. Service Response

After receiving the ciphertexts of n users, FD firstly disturbs these ciphertexts and then transmits them to CS. After receiving disturbed ciphertexts, CS first computes the geometric center of n candidate locations, and then calculates the Euclidean distances between the geometric center and each candidate position in the encrypted domain, finally sends n encrypted Euclidean distances to LDSP. LDSP obtains n Euclidean distances by decrypting the corresponding ciphertexts with its private key SK_{SP}, and then determines the minimum Euclidean distance and sends the corresponding index to FD. According to the index, FD returns the corresponding coordinates ciphertexts of the chosen location to LDSP. At last, LDSP utilizes a broadcast encryption scheme to re-encrypt the decrypted location, and then broadcasts the ciphertext to users. The detailed processing is as the following steps:

Step 3. After receiving n users' ciphertexts, FD first adds noise to C_{i1}, C_{i2}, C_{i3}, C_{i4}, C'_{i1}, C'_{i2}, and then respectively disturbs them according to three different random permutations.

Step 3.1. FD picks a random number s that satisfies $s^2 \cdot 4T^2 < N$ and calculates:

$$\bar{C}_{i1} = C_{i1}^{s^2} = g^{\bar{x}_i^2} \cdot \bar{r}_{i1}^N \bmod N^2 \quad (9)$$

$$\bar{C}_{i2} = C_{i2}^s = g^{(s \cdot N - \bar{x}_i)} \cdot \bar{r}_{i2}^N \bmod N^2 \quad (10)$$

$$\bar{C}_{i3} = C_{i3}^{s^2} = g^{\bar{y}_i^2} \cdot \bar{r}_{i3}^N \bmod N^2 \quad (11)$$

$$\bar{C}_{i4} = C_{i4}^s = g^{(s \cdot N - \bar{y}_i)} \cdot \bar{r}_{i4}^N \bmod N^2 \quad (12)$$

$$\bar{C}'_{i1} = {C'_{i1}}^s = g^{\bar{x}_i} \cdot \bar{r'}_{i1}^N \bmod N^2 \quad (13)$$

$$\bar{C}'_{i2} = {C'_{i2}}^s = g^{\bar{y}_i} \cdot \bar{r'}_{i2}^N \bmod N^2 \quad (14)$$

where $\bar{x}_i = s \cdot x_i$, $\bar{y}_i = s \cdot y_i$, $\bar{r}_{i1} = r_{i1}^{s^2}$, $\bar{r}_{i2} = r_{i2}^s$, $\bar{r}_{i3} = r_{i3}^{s^2}$, $\bar{r}_{i4} = r_{i4}^s$, $\bar{r}'_{i1} = {r'}^s_{i1}$, $\bar{r}'_{i2} = {r'}^s_{i2}$.

Step 3.2. FD randomly chooses three permutations of $\{1, 2, \cdots, n\}$: $\{p_1^{(1)}, p_2^{(1)}, \cdots, p_n^{(1)}\}$, $\{p_1^{(2)}, p_2^{(2)}, \cdots, p_n^{(2)}\}$ and $\{p_1^{(3)}, p_2^{(3)}, \cdots, p_n^{(3)}\}$.

According to the random permutation $\{p_1^{(1)}, p_2^{(1)}, \cdots, p_n^{(1)}\}$, FD shuffles $\{\bar{C}_1, \bar{C}_2, \cdots, \bar{C}_n\}$ to obtain $\{\bar{\bar{C}}_1, \bar{\bar{C}}_2, \cdots, \bar{\bar{C}}_n\}$ where $\bar{C}_i = \bar{C}_{i1}||\bar{C}_{i2}||\bar{C}_{i3}||\bar{C}_{i4}$, $\bar{\bar{C}}_i = \bar{C}_{p_i^{(1)}}$. In other words, $\bar{\bar{C}}_{i1} = \bar{C}_{{p_i^{(1)}}1}$, $\bar{\bar{C}}_{i2} = \bar{C}_{{p_i^{(1)}}2}$, $\bar{\bar{C}}_{i3} = \bar{C}_{{p_i^{(1)}}3}$, $\bar{\bar{C}}_{i4} = \bar{C}_{{p_i^{(1)}}4}$.

According to the random permutation $\{p_1^{(2)}, p_2^{(2)}, \cdots, p_n^{(2)}\}$, FD disturbs $\{\bar{C}'_{11}, \bar{C}'_{21}, \cdots, \bar{C}'_{n1}\}$ to obtain $\{\bar{\bar{C}}'_{11}, \bar{\bar{C}}'_{21}, \cdots, \bar{\bar{C}}'_{n1}\}$ where $\bar{\bar{C}}'_{i1} = \bar{C}'_{{p_i^{(2)}}1}$.

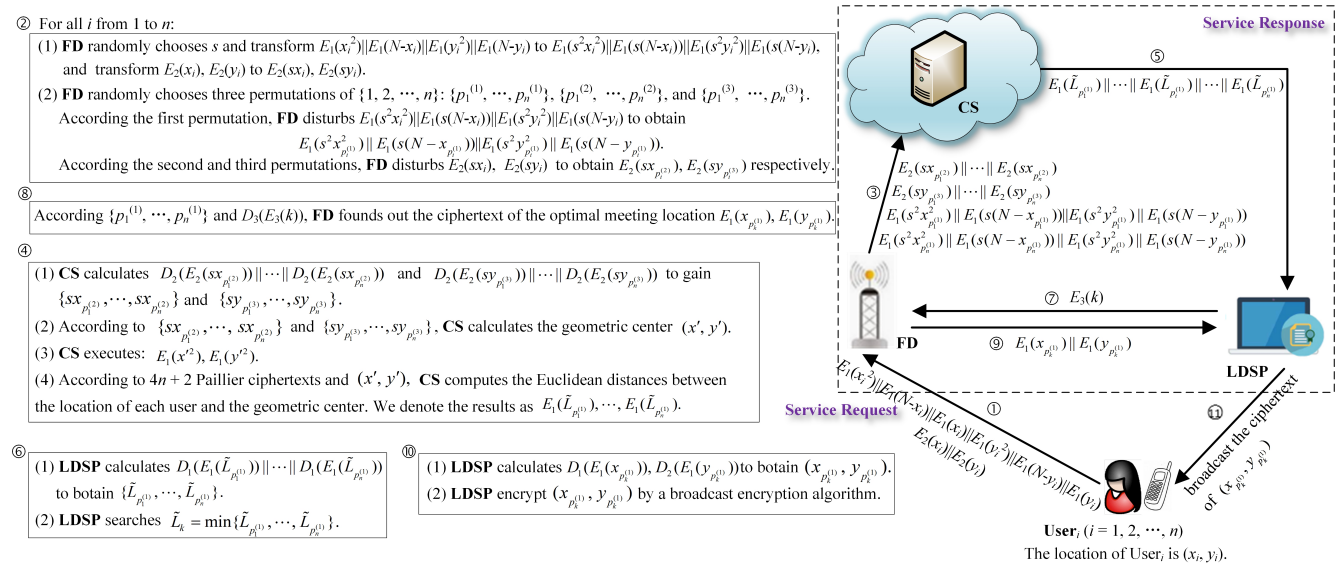

Fig. 3. The Prime Process Procedure of FMLD.

According to the random permutation $\{p_1^{(3)}, p_2^{(3)}, \cdots, p_n^{(3)}\}$, FD perturbs $\{\bar{C}'_{12}, \bar{C}'_{22}, \cdots, \bar{C}'_{n2}\}$ to obtain $\{\bar{\bar{C}}'_{12}, \bar{\bar{C}}'_{22}, \cdots, \bar{\bar{C}}'_{n2}\}$ where $\bar{C}'_{i2} = \bar{C}'_{p_i^{(3)}2}$.

FD sends $(\bar{\bar{C}}_1 \,||\, \bar{\bar{C}}_2 \,||\, \cdots \,||\, \bar{\bar{C}}_n, \bar{\bar{C}}'_{11} \,||\, \bar{\bar{C}}'_{21} \,||\, \cdots \,||\, \bar{\bar{C}}'_{n1}, \bar{\bar{C}}'_{12} \,||\, \bar{\bar{C}}'_{22} \,||\, \cdots \,||\, \bar{\bar{C}}'_{n2})$ to CS. Besides, FD stores C_{i5}, C_{i6}, and $\{p_1^{(1)}, p_2^{(1)}, \cdots, p_n^{(1)}\}$, and then removes $\{p_1^{(2)}, p_2^{(2)}, \cdots, p_n^{(2)}\}$, $\{p_1^{(3)}, p_2^{(3)}, \cdots, p_n^{(3)}\}$ and s.

Step 4. After receiving the ciphertexts from FD, CS achieves the computation of Euclidean distance as follows:

Step 4.1. CS decrypts $\bar{\bar{C}}'_{11} \,||\, \bar{\bar{C}}'_{21} \,||\, \cdots \,||\, \bar{\bar{C}}'_{n1}$ and $\bar{\bar{C}}'_{12} \,||\, \bar{\bar{C}}'_{22} \,||\, \cdots \,||\, \bar{\bar{C}}'_{n2}$ by carrying out the Paillier decryption algorithm under its private key SK_{CS}, in a result, obtains $\{\bar{\bar{x}}'_1, \bar{\bar{x}}'_2, \cdots, \bar{\bar{x}}'_n\}$ and $\{\bar{\bar{y}}'_1, \bar{\bar{y}}'_2, \cdots, \bar{\bar{y}}'_n\}$ where $\bar{\bar{x}}'_i = \bar{x}_{p_i^{(2)}} = s \cdot x_{p_i^{(2)}}$, $\bar{\bar{y}}'_i = \bar{y}_{p_i^{(3)}} = s \cdot y_{p_i^{(3)}}$.

Step 4.2. CS calculates the geometric center: $x' = (\bar{\bar{x}}'_1 + \bar{\bar{x}}'_2 + \cdots + \bar{\bar{x}}'_n)/n$ and $y' = (\bar{\bar{y}}'_1 + \bar{\bar{y}}'_2 + \cdots + \bar{\bar{y}}'_n)/n$.

Step 4.3. CS randomly chooses $\bar{r}_1, \bar{r}_2 \in \mathbb{Z}_N^*$ and encrypts x'^2, y'^2 with the LDSP's public key PK_{SP}:

$$\bar{C}_{x'^2} = g^{x'^2} \cdot \bar{r}_1^N \bmod N^2 \tag{15}$$

$$\bar{C}_{y'^2} = g^{y'^2} \cdot \bar{r}_2^N \bmod N^2 \tag{16}$$

Step 4.4. CS computes the Euclidean distances as follows:

$$\begin{aligned}\bar{\bar{C}}_{L_i} &= \bar{\bar{C}}_{i1} \cdot \bar{\bar{C}}_{i2}^{x'} \cdot \bar{C}_{x'^2} \cdot \bar{\bar{C}}_{i3} \cdot \bar{\bar{C}}_{i4}^{y'} \cdot \bar{C}_{y'^2} \\ &= g^{\bar{\bar{x}}_i{}^2 + (sN - \bar{\bar{x}}_i)\cdot x' + x'^2 + \bar{\bar{y}}_i{}^2 + (sN - \bar{\bar{y}}_i)\cdot y' + y'^2} \cdot R_i^N \bmod N^2 \\ &= g^{\bar{\bar{L}}_i} \cdot R_i^N \bmod N^2\end{aligned} \tag{17}$$

where $\bar{\bar{x}}_i = \bar{x}_{p_i^{(1)}} = s \cdot x_{p_i^{(1)}}$, $\bar{\bar{y}}_i = \bar{y}_{p_i^{(1)}} = s \cdot y_{p_i^{(1)}}$, $R_i = \bar{\bar{r}}_{i1} \cdot \bar{\bar{r}}_{i2}{}^{x'} \cdot \bar{r}_1 \cdot \bar{\bar{r}}_{i3} \cdot \bar{\bar{r}}_{i4}{}^{y'} \cdot \bar{r}_2 = \bar{r}_{p_i^{(1)}1} \cdot \bar{r}^{x'}_{p_i^{(1)}2} \cdot \bar{r}_1 \cdot \bar{r}_{p_i^{(1)}3} \cdot \bar{r}^{y'}_{p_i^{(1)}4} \cdot \bar{r}_2 = (r_{p_i^{(1)}1})^{s^2} \cdot (r_{p_i^{(1)}2})^{sx'} \cdot \bar{r}_1 \cdot (r_{p_i^{(1)}3})^{s^2} \cdot (r_{p_i^{(1)}4})^{sy'} \cdot \bar{r}_2$, $\bar{\bar{L}}_i = \bar{L}_{p_i^{(1)}} = s^2 \cdot L_{p_i^{(1)}}$, where $L_{p_i^{(1)}}$ represents the square of the Euclidean distance between the raw location $(x_{p_i^{(1)}}, y_{p_i^{(1)}})$ and the geometric center (x, y).

Step 4.5. CS transmits $\bar{\bar{C}}_{L_1} || \bar{\bar{C}}_{L_2} || \cdots || \bar{\bar{C}}_{L_n}$ to LDSP.

Step 5. Upon receiving the ciphertexts from CS, LDSP determines the appropriate meeting location as follows:

Step 5.1. LDSP decrypts the received ciphertexts by carrying out the Paillier decryption algorithm under its private key SK_{SP}, in a result, obtains $\bar{\bar{L}}_1, \bar{\bar{L}}_2, \cdots, \bar{\bar{L}}_n$.

Step 5.2. LDSP lookups the minimum value in $\{\bar{\bar{L}}_1, \bar{\bar{L}}_2, \cdots, \bar{\bar{L}}_n\}$ by adopting some kind of sorting algorithm or searching algorithm, assume the minimum is $\bar{\bar{L}}_k$ where $k \in \{1, 2, \cdots, n\}$.

Step 6. LDSP returns the response of users location service request as follows:

Step 6.1. LDSP encrypts the index k with PK_{FD} and sends the ciphertext to FD.

Step 6.2. Upon receiving the ciphertext, FD decrypts it with SK_{FD} to obtain the index k, and then checks the random permutation $\{p_1^{(1)}, p_2^{(1)}, \cdots, p_n^{(1)}\}$ with k to gain the corresponding value $p_k^{(1)}$. Note that $p_k^{(1)}$ may be equal to k, which has a probability of $\frac{1}{n}$. And then, FD sends $(C_{p_k^{(1)}5}, C_{p_k^{(1)}6})$ to LDSP.

Step 6.3. LDSP decrypts $(C_{p_k^{(1)}5}, C_{p_k^{(1)}6})$ by carrying out the Paillier decryption algorithm under its private key SK_{SP} to obtain the fair meeting location $(x_{p_k^{(1)}}, y_{p_k^{(1)}})$, and then LDSP encrypts $(x_{p_k^{(1)}}, y_{p_k^{(1)}})$ with the group users' public $pk_1, pk_2, \cdots, pk_n$ carrying out the broadcast encryption $BEnc()$ [25] and broadcasts the ciphertext to $\text{U}_1, \text{U}_2, \cdots, \text{U}_n$.

Instantiation: we leverage an example in Fig.4 to illustrate how our scheme FMLD works. To begin with, five users generate six Paillier ciphertexts with PK_{SP} and two Paillier ciphertexts with PK_{CS}, and then send these ciphertexs to FD. Upon receiving these ciphertexts, FD embeds noise to $\{C_1, C_2, C_3, C_4, C_5\}$, $\{C'_{11}, C'_{21}, C'_{31}, C'_{41}, C'_{51}\}$, $\{C'_{12}, C'_{22}, C'_{32}, C'_{42}, C'_{52}\}$ and shuffles them according to three random permutations respectively and sends them to CS. Here, we assume the three random permutations are $\{p_1^{(1)}, p_2^{(1)}, p_3^{(1)}, p_4^{(1)}, p_5^{(1)}\} = \{2, 4, 1, 5, 3\}$, $\{p_1^{(2)}, p_2^{(2)}, p_3^{(2)},$

$p_4^{(2)}$, $p_5^{(2)}\}$ = $\{4, 5, 1, 3, 2\}$, $\{p_1^{(3)}$, $p_2^{(3)}$, $p_3^{(3)}$, $p_4^{(3)}$, $p_5^{(3)}\}$ = $\{3, 5, 2, 1, 4\}$. Finally, FD stores (C_{15}, C_{16}), (C_{25}, C_{26}), (C_{35}, C_{36}), (C_{45}, C_{46}), (C_{55}, C_{56}), and $\{p_1^{(1)}, p_2^{(1)}, p_3^{(1)}, p_4^{(1)}$, $p_5^{(1)}\}$, and deletes the noise and $\{p_1^{(2)}, p_2^{(2)}, p_3^{(2)}, p_4^{(2)}, p_5^{(2)}\}$, $\{p_1^{(3)}, p_2^{(3)}, p_3^{(3)}, p_4^{(3)}, p_5^{(3)}\}$. Upon receiving the ciphertexts came from FD, CS first calculates the geometric center and then computes five Euclidean distances on Paillier ciphertexts, finally sends the results $\{\bar{\bar{C}}_{L_1}, \bar{\bar{C}}_{L_2}, \bar{\bar{C}}_{L_3}, \bar{\bar{C}}_{L_4}, \bar{\bar{C}}_{L_5}\}$ to LDSP. LDSP decrypts these ciphertexts with its private key SK_{SP}, and discovers $\bar{\bar{L}}_4$ is minimum, after that sends $k = 4$ to FD. Note the message sent here should be ciphertext of k, which is not shown in Fig.4 for the sake of simplicity. According to $k = 4$, FD checks the random permutation $\{p_1^{(1)} = 2, p_2^{(1)} = 4, p_3^{(1)} = 1, p_4^{(1)} = 5, p_5^{(1)} = 3\}$ and obtains $p_k^{(1)} = p_4^{(1)} = 5$, and then returns (C_{55}, C_{56}) to LDSP. After decrypting (C_{55}, C_{56}), LDSP broadcasts the chosen location by using the broadcast encryption $BEnc()$ [25].

VI. Correctness Analysis

In this section, we analyze whether the following equation holds: $AL_{FMLD}((x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)) = AL_{FMLD}((\bar{\bar{x}}_1, \bar{\bar{y}}_1), (\bar{\bar{x}}_2, \bar{\bar{y}}_2), \cdots, (\bar{\bar{x}}_n, \bar{\bar{y}}_n))$, where $\bar{\bar{x}}_i = \bar{x}_{p_i^{(1)}} = s \cdot x_{p_i^{(1)}}$, $\bar{\bar{y}}_i = \bar{y}_{p_i^{(1)}} = s \cdot y_{p_i^{(1)}}$ $(i = 1, 2, \cdots, n)$, $\{p_1^{(1)}$, $p_2^{(1)}$, $\cdots$, $p_n^{(1)}\}$ is a random permutation of $\{1, 2, \cdots, n\}$. According to **Step 4.1**, we have

$$\begin{aligned} x' &= s \cdot \frac{1}{n}\sum_{i=1}^{n} x_{p_i^{(2)}} = s \cdot \frac{1}{n}\sum_{i=1}^{n} x_i = s \cdot x \\ y' &= s \cdot \frac{1}{n}\sum_{i=1}^{n} y_{p_i^{(3)}} = s \cdot \frac{1}{n}\sum_{i=1}^{n} y_i = s \cdot y \end{aligned} \tag{18}$$

where $\{p_1^{(2)}, p_2^{(2)}, \cdots, p_n^{(2)}\}$ and $\{p_1^{(3)}, p_2^{(3)}, \cdots, p_n^{(3)}\}$ are two random permutations of $\{1, 2, \cdots, n\}$. According to Eq.17, we can obtain

$$\begin{aligned} \bar{\bar{L}}_i &= (\bar{\bar{x}}_i - x')^2 + (\bar{\bar{y}}_i - y')^2 = (\bar{x}_{p_i^{(1)}} - x')^2 + (\bar{y}_{p_i^{(1)}} - y')^2 \\ &= (s \cdot x_{p_i^{(1)}} - s \cdot x)^2 + (s \cdot y_{p_i^{(1)}} - s \cdot y)^2 \\ &= s^2 \cdot ((x_{p_i^{(1)}} - x)^2 + (y_{p_i^{(1)}} - y)^2) = s^2 \cdot L_{p_i^{(1)}} \end{aligned} \tag{19}$$

Assume $AL_{FMLD}((\bar{\bar{x}}_1, \bar{\bar{y}}_1), \cdots, (\bar{\bar{x}}_n, \bar{\bar{y}}_n)) = k$, that is $\bar{\bar{L}}_k = min\{\bar{\bar{L}}_1, \cdots, \bar{\bar{L}}_n\}$. We have

$$\begin{aligned} &\bar{\bar{L}}_k = s^2 \cdot L_{p_k^{(1)}} = min\{s^2 \cdot L_{p_1^{(1)}}, \cdots, s^2 \cdot L_{p_n^{(1)}}\} \\ &\Rightarrow L_{p_k^{(1)}} = min\{L_{p_1^{(1)}}, \cdots, L_{p_n^{(1)}}\} \\ &\Rightarrow L_k = min\{L_1, \cdots, L_n\} \end{aligned} \tag{20}$$

Therefore, $AL_{FMLD}((x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)) = k$. The equation $AL_{FMLD}((x_1, y_1), (x_2, y_2), \cdots, (x_n, y_n)) = AL_{FMLD}((\bar{\bar{x}}_1, \bar{\bar{y}}_1), (\bar{\bar{x}}_2, \bar{\bar{y}}_2), \cdots, (\bar{\bar{x}}_n, \bar{\bar{y}}_n))$ holds.

VII. Security Analysis

Following our design goals, we discuss how the presented scheme (FMLD) realizes the location privacy preservation of users.

Theorem 1: Passive attacks launched by internal adversaries can be resisted in the proposed scheme.

Proof: Since users first encrypt their location coordinates with the LDSP's public key PK_{SP} and the CS's public key PK_{CS} and then send these ciphertexts to FD, FD cannot obtain any user's location information without LDSP's private key SK_{SP} and the CS's private key SK_{CS}.

It is clear that CS cannot gain any user's location information from these ciphertexts encrypted by the LDSP's public key PK_{SP} sent from FD. CS can obtain n x-coordinates and n y-coordinates which have been perturbed according to different random permutations of $\{1, 2, \cdots, n\}$ and have been blinded by a random number s. Therefore, CS obtains nothing information about the location of users.

Uponing decrypting received ciphertexts, LDSP can gain n plaintexts each of which is s^2 times of the square of the Euclidean distance between one user's raw location and the geometric center. Besides, these ciphertexts have been perturbed based on a random permutation of $\{1, 2, \cdots, n\}$. Hence, LDSP gains nothing information about the locations of users through these inputs and the intermediate results. Finally, LDSP obtains the specific coordinate values of the fair meeting location through decrypting the ciphertext comes from FD, but LDSP cannot identify this location belongs to which user and only knows that the place belongs to one of the n users.

Consequently, FMLD achieves the location privacy preservation of users under passive attacks launched by internal adversaries.

Theorem 2: Active attacks launched by internal adversaries can be resisted in the proposed scheme.

Proof: In this attack model, CS and LDSP may share each other's inputs and intermediate results each other. LDSP can use its private key SK_{SP} to decrypt $(\bar{\bar{C}}_1 \parallel \bar{\bar{C}}_2 \parallel \cdots \parallel \bar{\bar{C}}_n)$ received by CS. Hence, LDSP and CS have the following knowledge: $((s \cdot x_{p_i^{(1)}})^2, -s \cdot x_{p_i^{(1)}}, (s \cdot y_{p_i^{(1)}})^2, -s \cdot y_{p_i^{(1)}})$, $s \cdot x_{p_i^{(2)}}$, $s \cdot y_{p_i^{(3)}}$, for all $i \in \{1, 2, \cdots, n\}$. LDSP and CS cannot gain the raw coordinate values of users' locations since unknowing the random number s and cannot identify $((s \cdot x_{p_i^{(1)}})^2, -s \cdot x_{p_i^{(1)}}$, $(s \cdot y_{p_i^{(1)}})^2$, $-s \cdot y_{p_i^{(1)}})$ belong to which user because $\{p_1^{(1)}$, $p_2^{(1)}$, $\cdots$, $p_n^{(1)}\}$ is a random permutation of $\{1, 2, \cdots, n\}$.

Some users also probably cooperate for obtaining other users' location information. The knowledge these users share is their location and the fair meeting location. Hence they try their best to guess whose position is the fair location. Assume there are n' $(n' < n)$ users conspire together. If the fair meeting location belongs to one of these conspiring users, they cannot obtain anything information about other users' location; otherwise, they only know the fair meeting location belongs to one of the other $n - n'$ users. Conceivably, the probability that these conspiring users successfully guess is $\frac{n-n'}{n} \cdot \frac{1}{n-n'} = \frac{1}{n}$. Therefore, the users' collusion cannot help them gain any information about other users' locations.

In summary, FMLD achieves the location privacy preservation of users under active attacks launched by internal adversaries.

Theorem 3: Confidentiality of users' locations against external adversaries can be guaranteed in the proposed scheme.

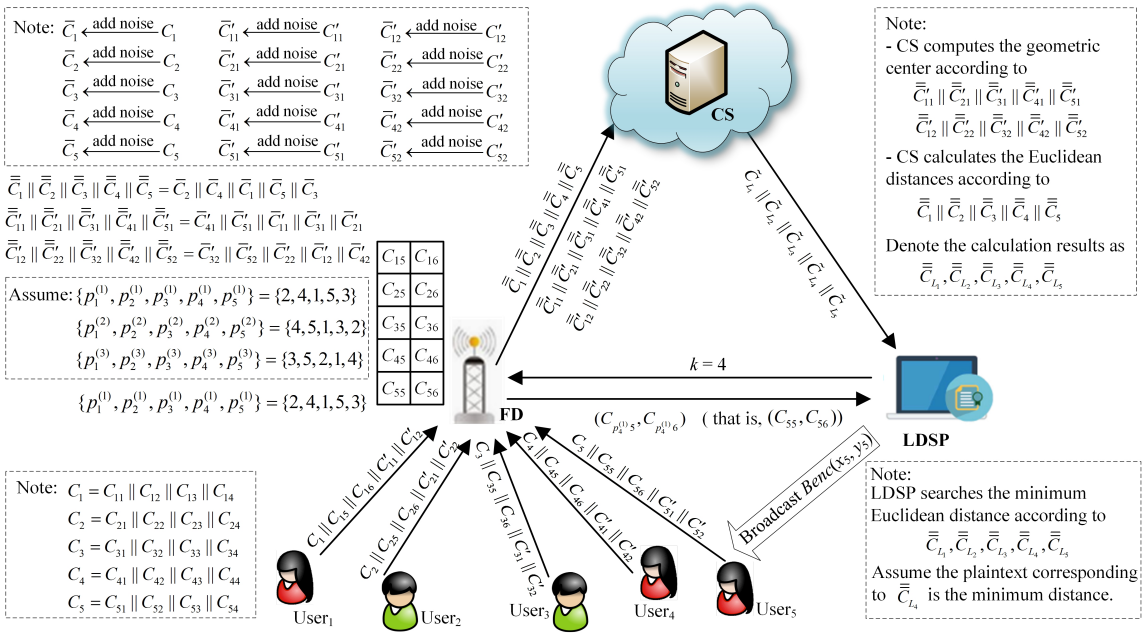

Fig. 4. An Instantiation of FMLD.

Proof: As users' location coordinates are encrypted with the LDSP's public key PK_{SP} and the CS's public key PK_{CS}, there is a confidentiality guarantee against eavesdropping communication and intruding in the databases by external adversaries.

VIII. Performance Evaluation

In this section, we study the results of controlled experiments and compare the performance of our scheme FMLD with the ElGamal-Paillier-based PPFRVP scheme [7] and the ORFMLD scheme [8] in terms of computation and communication costs. FMLD adopts Paillier cryptosystem, PPFRVR uses Paillier and ElGamal cryptosystems, and ORFMLD utilizes BGN and ElGamal cryptosystems.

A. Computation Costs Analysis

The process of the encryption and decryption of Paillier, BGN, ElGamal cryptosystems can be divided into some basic computations: the exponentiation, multiplication operations in $\mathbb{Z}^*_{N^2}$, $\mathbb{Z}^*_N$ and $\mathbb{Z}^*_{p'}$, respectively, and Pollard's lambda method. Note that p' is a large prime and the system parameter of ElGamal. We set $|N| = |p'|$, so the computation time of one exponentiation operation in $\mathbb{Z}^*_N$ is equal that of one exponentiation in $\mathbb{Z}^*_{p'}$. We use T_{e_1} to denote the computational time of an exponentiation operation in $\mathbb{Z}^*_N$, and denote the computational times of an exponentiation operation in $\mathbb{Z}^*_{N^2}$, an multiplication operation in $\mathbb{Z}^*_N$ and that in $\mathbb{Z}^*_{N^2}$, a multiplication in $\mathbb{G}$, and a pairing operation by $T_{e_2}, T_{m_1}, T_{m_2}, T_m, T_b$, respectively. And, we denote the computation time of using Pollard's lambda method to compute the discrete logarithm by T_p.

Computation cost of PPERVP: Each user carries out $n+1$ times Paillier encryption operations, 2 times ElGamal encryption operations, n times Pailliers decryption operations, and $n-1$ ElGamal decryption operations. Therefore, each user's computational overhead is $(3n+2)T_{e_2}+(2n+1)T_{m_2}+(n+3)T_{e_1}+(n+1)T_{m_1}$, the total user-side computation cost is $n(3n+2)T_{e_2}+n(2n+1)T_{m_2}+n(n+3)T_{e_1}+n(n+1)T_{m_1}$. The server takes $\frac{n(n-1)}{2}*4=2n(n-1)$ times modular multiplication operations on ElGamal ciphertexts, and $\frac{n(n-1)}{2}*5$ times modular multiplication operations and $\frac{n(n-1)}{2}*2=n(n-1)$ times modular exponential operations on Paillier ciphertexts. Therefore, the total server-side computation cost is $2n(n-1)T_{m_1}+\frac{5n(n-1)}{2}T_{m_2}+n(n-1)T_{e_2}$.

Computation cost of ORFMLD: Each user carries out 3 times BGN encryption operations and 3 ElGamal encryption operations. Therefore, each user's computational overhead is $12T_{e_1}+3T_m+3T_{m_1}$, the total user-side computation cost is $12nT_{e1}+3nT_m+3nT_{m1}$. CC caries out $3n$ times ElGamal decryption operations, n times BGN encryption operations, and $\frac{n(n-1)}{2}*4=2n(n-1)$ times bilinear map operations. MLDS takes $\frac{n(n-1)}{2}$ times BGN decryption operations. Therefore, the total server-side computation cost is $\frac{n(n+9)}{2}T_{e_1}+3nT_{m1}+nT_m+2n(n-1)T_b+\frac{n(n-1)}{2}T_p$.

Computation cost of FMLD: Each user carries out 8 times Paillier encryption operations. Therefore, each user's computational overhead is $16T_{e_2}+8T_{m_2}$, the total user-side computation cost is $16nT_{e2}+8nT_{m2}$. FD takes $6n$ times modular exponential operations on Paillier ciphertexts and one time Paillier decryption operation. CS caries out $2n$ Paillier decryption operations, 2 times Paillier encryption operations, and $5n$ times modular multiplication operations on Paillier ciphertexts. LDSP executes n times Paillier decryption operations and one time Paillier encryption operation. Therefore, the total server-side computation cost is $(9n+7)T_{e_2}+(8n+4)T_{m_2}$.

Since the multiplications in $\mathbb{Z}^*_{N^2}$, $\mathbb{Z}^*_N$ and $\mathbb{Z}^*_{p'}$ are negligibly small compared to the exponentiation and pairing operations, in this paper, the computational cost of these multiplication

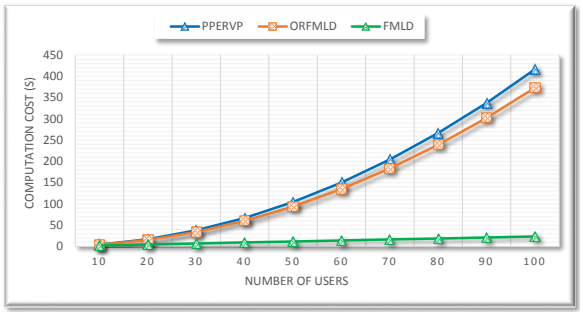

Fig. 5. Comparison of Computation Costs.

TABLE III
RESPONSE TIME OF FMLD

User Number	10	20	30	40	50
Response Time (S)	0.91	1.76	2.61	3.46	4.31
User Number	60	70	80	90	100
Response Time (S)	5.15	6.00	6.85	7.70	8.55

operations is negligible. In other words, we omit T_{m_1} and T_{m_2} in the comparison analysis of computation costs. Table II shows the comparison of the computation overhead. T_{e_1}, T_{e_2}, T_m, T_b, and T_p are constants, and the relationships between them are constant multiples. Therefore, according to Table II, it is clear that, along with increasing users' number, the computation cost growth rate of PPERVP is $O(n^2)$, that of ORFMLD is $O(n^2)$, and that of our scheme FMLD is $O(n)$. In other words, FMLD is more efficient in terms of computation cost.

To further illustrate this, we utilize JPBC Library [26] to conduct our experiment on a 3.10-GHz Inter Pentium G3240 processor, 4GB RAM, and 64-bit operating system, computing machine. For Paillier and ElGamal encryptions, we adopt 1024-bit secret keys. For BGN encryption, we choose 160-bit secret key and assume the message space consists of integers in the set $\{0, 1, \cdots, T\}$, where $|T| << 512$, the expected time is around $O(\sqrt{T})$ when using the Pollard's lambda method [27], here we let $|T| = 13$. Specifically, we have the result $T_{e_1} = 3.66$ms, $T_m = 0.06$ms, $T_{e_2} = 9.37$ms, $T_b = 17.65$ms. When $T_p = 0.02$ms. Based on Table II, we depict the variation of computational costs of n in Fig.5. From Fig.5, it is clear that the computation cost of FMLD grows linearly with the number of users, and the growth of computation cost of FMLD is slower than that of PPERVP and ORFMLD.

Moreover, because there is no interaction between users and servers in response processing of FMLD, we can take the total server-side computation cost as the response time of service. Table III shows some response times of FMLD. From Table III, we can find when there are 100 users to require determining a meeting location, the response time is about 8.55 seconds, which is marginally tolerable for users. When there are 40 users, the response time is about 3.46 seconds, which is terrific for users. In FMLD, the computation cost of each user is about 1.5 seconds, which is suitable for resource-restricted mobile devices. Hence, our FMLD scheme is efficient and practical.

TABLE IV
COMMUNICATION OVERHEADS OF PPEPVP, ORFMLD, AND FMLD

PPERVP [7]	User→LDS(1st) $n(2L_P + 4L_E)$	LDS→User $2n(n-1)L_E$	User→LDS(2nd) $n(n-1)L_P$	
ORFMLD [8]	User→CC $n(3L_B + 6L_E)$	C→MLDS $n(n-1)L_B$		
FMLD	User→FD $n(6L_P + 4L_E)$	FD→CS $4n(L_P + L_E)$	CS→LDSP nL_P	LDSP→FD $1L_E$

TABLE V
COMPARISON OF COMMUNICATION OVERHEAD

PPERVP [7]	ORFMLD [8]	FMLD
$(n^2+n)(L_P+2L_E)$	$(n^2+2n)L_B+6nL_E$	$11nL_P+(8n+1)L_E$

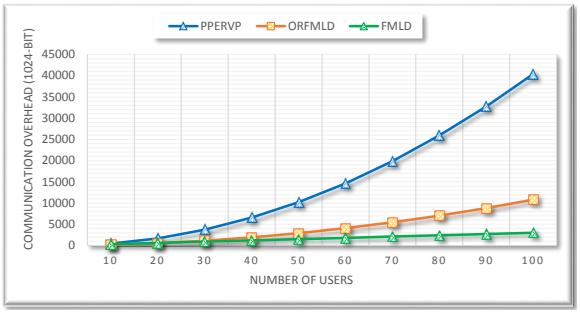

Fig. 6. Comparison of Commutation Overheads.

B. Communication Overhead Analysis

We set the security parameter $\kappa = 512$ for Paillier and BGN encryptions and set the security parameter $\kappa' = 1024$ for ElGamal encryption. Therefore, the length of a Paillier ciphertext is $L_P = 2048$ bits, that of a BGN ciphertext is $L_B = 1024$ bits, and that of an ElGamal ciphertext is $L_E = 1024$ bits. Communication overheads of PPERVP, ORFMLD, and FMLD are shown in Table IV. The total communication overhead comparison of the three schemes is shown in Table V. Because L_P, L_B, and L_E are constants, according to Table V, the growing rates of communication overhead of PPERVP and ORFMLD along with increasing users' number are $O(n^2)$, that of our scheme FMLD is $O(n)$. Therefore, FMLD is more efficient in terms of communication overhead. Fig.6 demonstrates the overall communication overhead of PPERVP, ORFMLD, and FMLD. From Fig.6, it is clear that the communication overhead of FMLD is close to that of ORFMLD, but along with the increasing number of users, the advantage in terms of communication cost of FMLD are more visible.

IX. CONCLUSION

In this paper, we propose a novel solution to determinate a fair meeting location. In our method, there are only n Euclidean distances need to be calculated. To realize our approach without disclosing n user's location privacy and consider the limited resource of mobile devices, we propose a lightweight privacy-preserving fair meeting location determination scheme (namely FMLD). We synergistically employ Paillier homomorphic encryption and random permutation method to achieve FMLD. FMLD has linear computation and

TABLE II
COMPARISON OF THE COMPUTATION COST

	User-side Computation Cost	Server-side Computation Cost	Total Computation Cost
PPERVP [7]	$(3n^2+2n)T_{e_2}+(n^2+3n)T_{e_1}$	$(n^2-n)T_{e_2}$	$(4n^2+n)T_{e_2}+(3n^2+n)T_{e_1}$
ORFMLD [8]	$12nT_{e1}+3nT_m$	$\frac{n(n+9)}{2}T_{e_1}+nT_m+2n(n-1)T_b+\frac{n(n-1)}{2}T_p$	$\frac{n(n+33)}{2}T_{e_1}+4nT_m+2n(n-1)T_b+\frac{n(n-1)}{2}T_p$
FMLD	$16nT_{e2}+8nT_{m2}$	$(9n+7)T_{e_2}+(8n+4)T_{m_2}$	$(25n+7)T_{e_2}+(16n+4)T_{m_2}$

communication complexity, which thanks to FMLD only needs to calculate n Euclidean distances securely. The security analysis confirms FMLD's security properties. The security analysis confirms FMLD's security properties. Note that, FMLD is for a static application scene. In other words, we do not consider users' movement after submitting their candidate locations. Different movement speeds of users can lead the fair meeting location to be changed. Future research will discuss how to extend the current scheme to be suitable for a dynamic application scene.

REFERENCES

[1] X. Zhang, H. Huang, S. Huang, Q. Chen, T. Ju, and X. Du, "A context-aware location differential perturbation scheme for privacy-aware users in mobile environment," *Wireless Communications and Mobile Computing*, vol. 2018, 2018.

[2] H. Li, H. Zhu, S. Du, X. Liang, and X. S. Shen, "Privacy leakage of location sharing in mobile social networks: Attacks and defense," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 4, pp. 646–660, 2018.

[3] 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.

[4] G. Sun, Y. Xie, D. Liao, H. Yu, and V. Chang, "User-defined privacy location-sharing system in mobile online social networks," *Journal of Network and Computer Applications*, vol. 86, pp. 34–45, 2017.

[5] D. Song, J. Sim, K. Park, and M. Song, "A privacy-preserving continuous location monitoring system for location-based services," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 815613, 2015.

[6] S. Zhang, G. Wang, M. Z. A. Bhuiyan, and Q. Liu, "A dual privacy preserving scheme in continuous location-based services," *IEEE Internet of Things Journal*, vol. 5, no. 5, pp. 4191–4200, 2018.

[7] I. Bilogrevic, M. Jadliwala, V. Joneja, K. Kalkan, J.-P. Hubaux, and I. Aad, "Privacy-preserving optimal meeting location determination on mobile devices," *IEEE transactions on information forensics and security*, vol. 9, no. 7, pp. 1141–1156, 2014.

[8] X. Wang, "One-round secure fair meeting location determination based on homomorphic encryption," *Information Sciences*, vol. 372, pp. 758–772, 2016.

[9] G. Sun, V. Chang, M. Ramachandran, Z. Sun, G. Li, H. Yu, and D. Liao, "Efficient location privacy algorithm for internet of things (iot) services and applications," *Journal of Network and Computer Applications*, vol. 89, pp. 3–13, 2017.

[10] P. Research, "A survey by pyramid research," http://www.pyramidresearch.com/store/Report-Location-Based-Services.htm.

[11] X. Zhu, H. Chi, B. Niu, W. Zhang, Z. Li, and H. Li, "Mobicache: When k-anonymity meets cache," in *2013 IEEE Global Communications Conference (GLOBECOM)*. IEEE, 2013, pp. 820–825.

[12] G. Sun, D. Liao, H. Li, H. Yu, and V. Chang, "L2p2: A location-label based approach for privacy preserving in lbs," *Future Generation Computer Systems*, vol. 74, pp. 375–384, 2017.

[13] J. Lin, J. Niu, H. Li, and M. Atiquzzaman, "A secure and efficient location-based service scheme for smart transportation," *Future Generation Computer Systems*, vol. 92, pp. 694–704, 2019.

[14] M. Zhang, Y. Zhang, Y. Jiang, and J. Shen, "Obfuscating eves algorithm and its application in fair electronic transactions in public clouds," *IEEE Systems Journal*, 2019.

[15] M. Duckham and L. Kulik, "Location privacy and location-aware computing," in *Dynamic and Mobile GIS*. CRC press, 2006, pp. 63–80.

[16] C. Bettini, S. Mascetti, X. S. Wang, and S. Jajodia, "Anonymity in location-based services: Towards a general framework," in *2007 International Conference on Mobile Data Management*. IEEE, 2007, pp. 69–76.

[17] J.-D. Zhang and C.-Y. Chow, "Real: A reciprocal protocol for location privacy in wireless sensor networks," *IEEE Transactions on Dependable and Secure Computing*, vol. 12, no. 4, pp. 458–471, 2014.

[18] H. Wang, H. Huang, Y. Qin, Y. Wang, and M. Wu, "Efficient location privacy-preserving k-anonymity method based on the credible chain," *ISPRS International Journal of Geo-Information*, vol. 6, no. 6, p. 163, 2017.

[19] B. Niu, Q. Li, X. Zhu, G. Cao, and H. Li, "Enhancing privacy through caching in location-based services," in *2015 IEEE conference on computer communications (INFOCOM)*. IEEE, 2015, pp. 1017–1025.

[20] L. Calderoni, P. Palmieri, and D. Maio, "Location privacy without mutual trust: The spatial bloom filter," *Computer Communications*, vol. 68, pp. 4–16, 2015.

[21] T. Peng, Q. Liu, G. Wang, Y. Xiang, and S. Chen, "Multidimensional privacy preservation in location-based services," *Future Generation Computer Systems*, vol. 93, pp. 312–326, 2019.

[22] B. Mu and S. Bakiras, "Private proximity detection for convex polygons," *Tsinghua Science and Technology*, vol. 21, no. 3, pp. 270–280, 2016.

[23] S. M. Oteafy and H. S. Hassanein, "Iot in the fog: A roadmap for data-centric iot development," *IEEE Communications Magazine*, vol. 56, no. 3, pp. 157–163, 2018.

[24] H. Shen, M. Zhang, and J. Shen, "Efficient privacy-preserving cube-data aggregation scheme for smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1369–1381, 2017.

[25] D.-H. Phan, D. Pointcheval, S. F. Shahandashti, and M. Strefler, "Adaptive cca broadcast encryption with constant-size secret keys and ciphertexts," *International journal of information security*, vol. 12, no. 4, pp. 251–265, 2013.

[26] A. De Caro and V. Iovino, "Jpbc: Java pairing based cryptography," in *Proceedings of the 16th IEEE Symposium on Computers and Communications, ISCC 2011*, Kerkyra, Corfu, Greece, June 28 - July 1, 2011, pp. 850–855.

[27] J. M. Pollard, "Kangaroos, monopoly and discrete logarithms," *Journal of cryptology*, vol. 13, no. 4, pp. 437–447, 2000.

Hua Shen received her Ph.D. degree in computer science from the School of Computer Science, Wuhan University, China, in 2014. She is currently an Associate Professor in Hubei University of Technology. Her research interests include the technology of privacy-preserving, information security, and secure cloud computing.

Mingwu Zhang was a JSPS Fellow (Japan Society for the Promotion of Science) with the Institute of Mathematics for Industry, Kyushu University, Japan, from 2010 to 2012. From 2015 to 2016, he was a Senior Research Fellow with the Centre for Computer and Information Security, University of Wollongong, Australia. He is currently a Professor in Hubei University of Technology. His current research interests include cryptography technology for clouds and big data, and privacy preservation.

Hao Wang received his Ph.D. degree in computer science from Shandong University, China, in 2012. He is currently an Associate Professor in Shandong Normal University. His primary interest is public key cryptography, in particular, designing cryptographic primitives and provable security. At present, he is focusing on attribute-based cryptography, secure multi-party computation, and blockchain.

Fuchun Guo received the BS and MS degrees from Fujian Normal University China, in 2005 and 2008, respectively, and the PhD degree from the University of Wollongong, Australia in 2013. He is currently an associate research fellow at the School of Computing and Information Technology, University of Wollongong. His primary research interest is the public-key cryptography; in particular, protocols, encryption and signature schemes, and security proof.

Willy Susilo received a Ph.D. degree in Computer Science from University of Wollongong, Australia. He is a Professor and the Head of School of Computing and Information Technology and the director of Institute of Cybersecurity and Cryptology (iC^2) at the University of Wollongong. He was previously awarded the prestigious ARC Future Fellow by the Australian Research Council (ARC). His main research interests is applied cryptography. He is a senior member of IEEE.