IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 19, Issue 4, Ver. V. (Jul.-Aug. 2017), PP 19-24
www.iosrjournals.org

Secure Fair Rendez- Vous Point technique for minimizing meeting location on mobile device.

Mekala.Anuradha[1], Atchala.Sudhakar [2]
[1]M.Tech.IV Sem.(CSE) Lakireddy Balireddy College of Engineering,Mylavaram
[2]Asst. Professor, CSE Department, Lakireddy Balireddy College of Engineering, Mylavaram
Corresponding Author: Mekala.Anuradha[1]

Abstract: *present days people of urban and rural are using smart phones and mobile devices intensively. In Particular urban population depends on the applications and gadgets which are provided by the mobile devices and Smart phones to plan their daily life. The applications which are built on these devices mainly depend on the current or preferred locations of the user to provide the services they wish, which may cause damage to the privacy of mobile device users. In general no user wish to reveal their present location or the location they wish to go. We proposed pp algorithms which will provide an optimal location for group of users.*

Keywords*: Mobile devices, applications, privacy preserving.*

Date of Submission: 07-07-2017 Date of acceptance: 05-08-2017

I. Introduction

In urban areas due to the rapid development of smart phone technology made the people to use location based services on their mobile devices. Advantage has been taken by the service providers by providing ever growing location based services for mobile device users. Millions of people are using location based services (LBS), to get information of particular location [1]. The two features that are popularly used based on location services are location check-ins and location sharing. Using location checking, user can share his/her current location to family, friends etc..,or user can obtain location specific information from third party service provider. The other LBS services provide the location sharing by the group or number of users also becoming popular now-a-days. Almost 20% of mobile users are using location sharing services according to recent survey [4]. One of the most popular applications of such type is taxi sharing application. By using such applications user current and preferred locations can be known by service provider which may leads to bad consequences on user's financial, social, business and political status.

User's current location and preferred locations should be kept secretly from other participant user and third party service provider which is an important aspect in such LSB applications. If such information like users and their availabilities [7], are de-anonymized to known the preferences. The third party service provider can identify the user location current and preferred location pairs easily if the user is using service provider application very often. Even third party service will track the user details to provide the quality service can indirectly harm the privacy of the user if the details are leaked in unauthorized way.

In this work, we focus on particular problem called Fair Rendez-Vous point problem which is an issue in LSBSs. By using the set of user location preferences from the user, the FRVP problem will determine the location from the proposed location so that maximum distance between determined location and all the other preferred locations can be minimized that means it is fair to all users. Main goal of this paper is to provide privacy preserving practical techniques to solve the problem of FRVP, so that both the third party service provider and users who are participating cannot know locations of other users. Participating users can only know the optimal location.

We are going to solve the privacy problem of the user first by formulating the problem of FRVP as an problem of optimization, particularly the k-centre problem [12], and then privacy is provided among the participants with respect to one another and a third party service provider. Algorithms proposed by us will take the advantage of homomorphic properties of cryptosystems to compute an optimal fair rendez-vous point by using set of location preferences from the user. We provide an accurate analysis to show that our algorithms will not provide any way of guessing the participant preferred location. Including the theoretical analysis, we also made evaluation of practical efficiency and proposed algorithms performance by using the implementation of prototype on Nokia mobile device test beds. Finally we also propose the case of multi-preferences of the user based on priorities of location. We show the difference mainly in terms of performance and privacy, by using single preference case and initial experimental results are shown for the implementation of multi-preference.

II. System Architecture

We consider a system composed of two main entities: (i) a set of users (or mobile devices) $\mathbb{U} = \{u_1, . . . , u_N \}$ and (ii) a third-party service provider, called *Location Determination Server (LDS)*, which is responsible for privately computing the fair rendez-vous location or point from a set of userpreferred rendez-vous locations. Each user's mobile device is able to communicate with the LDS by means of some fixed infrastructure-based Internet connection.

Each user *ui* has the means to determine the coordinates $L_i = (x_i , y_i) \in \mathbb{N}^2$ of his preferred rendez-vous location. Let us consider a two-dimensional coordinate system, we proposed schemes are general enough and can be easily expand to other higher dimensional coordinate systems [14]. Users can either use their current position as their preferred rendez-vous location or they can specify some other preferred location (e.g., a point-of-interest such as a known restaurant) away from their current position. Users determine their current position (or positions of known points-of-interest) by using a positioning service, such as Global Positioning System or GPS. We assume that the positioning service is fairly accurate. GPS, for example, has an average positioning error between 3 and 7.8 meters.

We would like the readers to note that the goal of the positioning service is only to enable users to determine or select their preferred location, and that it should not be confused with the LDS. Users can continue to use the service of the LDS for privately computing the fair rendezvous location without using the positioning service, say by manually estimating their preferred rendez-vous location. A positioning service, if used, can continuously track users based on the positioning requests or it can behave maliciously and provide incorrect position information (or position information with large errors) to the users. In this work, we do not consider these adversarial scenarios involving the positioning service as these are orthogonal to the privacy preserving FRVP problem. In order to limit the information that the positioning service learns about the users' location requests, a private information retrieval technique [15] can be used. Moreover, a secure positioning system can be used to overcome the problem of cheating within the positioning service.

We define the set of the preferred rendez-vous locations of all users as $\mathbb{L} = \{L_i\}^{N}_{i=1}$. For the sake of simplicity, we consider line-of-sight Euclidean distances between preferred rendez-vous locations. Even though the actual real-world distance (road, railway, boat, etc.) between two locations is at least as large as their Euclidean distance, the proportion between distances in the real world is assumed to be correlated with the respective Euclidean distances.

The mobile devices are able to perform public-key cryptographic operations.We assume that each of the N users has his own public/private key pair (K_P^{ui}, K_s^{ui}), certified by a trusted CA, which is used to digitally sign/verify the messages that are sent to the LDS. Moreover, we assume that the N users share a common secret that is utilized to generate a shared public/private key pair (K_P^{Mv}, K_s^{Mv}) in an online fashion for each meeting setup instance v. The private key K_s^{Mv} Generated in this way is known only to all meeting participants, whereas the public key K_P^{Mv} is known to everyone including the LDS.

The LDS executes the Fair point rendez-vous algorithm on the inputs it receives from the users in order to compute the FRV point. The LDS is also perform public-key cryptographic functions. For instance, a common public-key infrastructure using the RSA cryptosystem could be employed. Let K_P^{LDS} be the public key, certified by a trusted CA, and K_s^{LDS} the corresponding private key of the LDS. K_P^{LDS} is publicly known and users encrypt their input to the FRVP algorithm using this key; the encrypted input can be decrypted by the LDS using its private key K_S^{LDS}. This ensures message confidentiality and integrity. For simplicity, we do not explicitly show the cryptographic operations involving LDS's public/private key.

A. Threat Model

1) Location Determination Server: The primary type of LDS adversarial behavior that we want to protect against is an *honest-but-curious* or semi-honest adversary, where the LDS is assumed to execute the algorithms correctly, i.e., take all the inputs and produce the output according to the algorithm, but is not fully trusted. It may try to learn information about the users' location preferences from the received inputs, the intermediate results and the produced outputs. In most practical settings, where service providers have a commercial interest in providing a faithful service to their customers, the assumption of a semi-honest LDS is generally sufficient. Given this goal of protecting against a semi-honest LDS, we will later also analyze how our proposed solutions fair against certain *active attacks*, including collusion with users and fake user generation.

2) Users: our main goal is to protect against semi-honest participating users who may want to learn the private location preferences of other users from the intermediate results and the output of the Fair point rendez-vous algorithm. We refer to such attacks as *passive attacks*. As user inputs are encrypted with the LDS's public key K_p^{LDS}, there is a confidentiality guarantee against basic eavesdropping by participants and non-participants. Given this goal of protecting against semi-honest users, we will later also analyze how our proposed solutions fair against certain *active attacks*, including collusion among users and input manipulation attacks.

III. PPFRVP Problem Formulation

In this work, we consider the problem of finding a rendezvous point among a set of user-proposed locations, such that (i) the rendez-vous point is *fair* with respect to the given input locations, (ii) each user learns only the final rendez-vous location and (iii) no participating user or third-party server learns private location preference of any other user involved in the computation. We refer to an algorithm that solves this problem as *Privacy-Preserving Fair Rendez-Vous Point (PPFRVP)* algorithm. In general, any PPFRVP algorithm A should accept the inputs and produce the outputs, as described below.

- *Input*: transformation f of private locations $L_i : f(L_1) \| f(L_2) \| \ldots \| f(L_N)$. where f is a secret-key based encryption function such that it is hard (success with only a negligible probability) to determine the input L_i without knowing the secret key, by just observing $f(L_i)$.
- *Output*: an output $f(L_{fair}) = g(f(L_1), \ldots, f(L_N))$, where g is a fairness function and $L_{fair} = (x_l, y_l) \in \mathbb{N}^2$ is the fair rendez-vous location such that it is hard for the LDS to determine L_{fair} by just observing $f(L_{fair})$. Given $f(L_{fair})$, each user should be able to compute $L_{fair} = f^{-1}(f(L_{fair}))$ by using a decryption routine and the shared secret key.

Fig. 1 shows a functional diagram of the PPFRVP protocol, where the PPFRVP algorithm A is executed by an LDS. The fairness function g can be defined in several ways, depending on the preferences of users or policies. Fig. 2 shows one such fairness function that minimizes the maximum displacement of any user to all other locations. This function is globally fair and can be easily extended to include additional constraints and parameters.

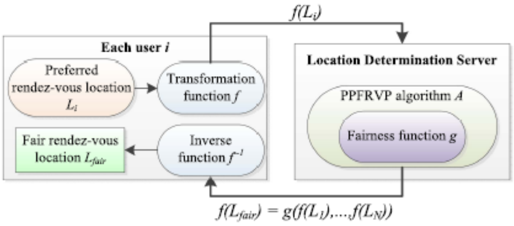
$f(L_i)$
Each user i
Preferred rendez-vous location L_i
Transformation function f
Fair rendez-vous location L_{fair}
Inverse function f^{-1}
Location Determination Server
PPFRVP algorithm A
Fairness function g
$f(L_{fair}) = g(f(L_1),...,f(L_N))$

Fig. 1. Functional diagram of the PPFRVP protocol.

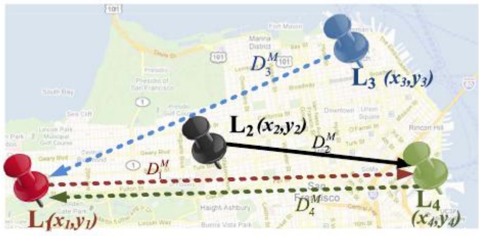

Fig. 2. PPFRVP scenario, where the fairness function is $g = \text{argmin}_i(D_i^M)$. The dashed arrows represent the maximum distance D_i^M from each user u_i to any user $j \neq i$, whereas the solid line is the minimum of all such maximum distances. The fair rendez-vous location is $L_{fair} = L_2 = (x_2, y_2)$.

IV. Proposed Solution To Ppfrvp Problem

In our paper, we outline the details of our proposed protocol for solving the Privacy preservingFPRV problem. In order to separate the optimization aspect from the implementation, we first formally outline the fairness and transformation functions and then discuss the construction of the PPFRVP protocol.

A. Fairness Function g

In order to determine a rendez-vous location that is *fair* to all users, the fairness function needs to optimize based on the spatial constraints set by the users' preferred locations. For example, a rendez-vous location $L_{fair} = (x_l , y_l)$ among N users $\mathbb{U} = \{ui\}^{N}_{i=1}$ will be fair to all users if everyone can reach L_{fair} in a "reasonable" amount of time. Another criterion is to minimize the total displacement of all users in order to reach L_{fair}, or simply, making sure that no user is "too far" from L_{fair} as compared to other users. We model the fairness criterion of the PPFRVP problem by using a formulation of the *k-center* problem. In the k-center problem, the goal is to determine k locations $(L_1, . . . , L_k)$ for placing facilities, among N possible candidates, such that the maximum distance from any place to its closest facility is minimized. For a two dimensional coordinate system, the Euclidean distance metric is usually employed.

As the PPFRVP problem is to determine a *single* fair rendezvous location from a set of user-preferred locations, we focus on the *k-center* formulation of the problem with $k = 1$. This choice is also grounded on the fact that not choosing L_{fair} from one of the location preferences $L_1, . . . , L_N$ might potentially result in a location L_{fair} that is not suited for the kind of meeting that the participants require. The solution can easily be extended or integrated with mapping applications (on the users' devices) so that POIs around L_{fair} are automatically suggested for the meeting. Fig. 2 shows a PPFRVP scenario modeled as a k-center problem. It should be noted that the current k-center formulation does not encompass other fairness parameters, such as accessibility of a place and the means of transportation. Later, we will extend our model to encompass multiple and prioritized user location preferences. Let $d_{i\,j} \geq 0$ be the Euclidean distance between two points L_i , $L_j \in \mathbb{N}^2$, and $D^M{}_i = \max j \neq i\ d_{ij}$ be the maximum distance from L_i to any other point L_j. The PPFRVP problem can be formally defined as follows.

Definition 1: The PPFRVP *problem is to privately compute a location* $L_{fair} \in \text{L} = \{L_1, . . . , L_N\}$, *where* $fair = arg\ \min_i D_i^M$.

B. Transformation Functions f

The fairness criteria g requires the computation of two functions on the user-preferred locations L_i : (i) the distance between any two locations L_i and L_j , $L_i \neq L_j$ and (ii) the minimum of the maximum of these distances. In order to solve the FRVP problem privately, we rely on computationally secure cryptographic primitives. We are interested in using cryptographic schemes that allow us to obliviously compute the Euclidean distance between two points and the maximization/minimization functions. We utilize cryptographic schemes with homomorphic properties as the transformation function f in our PPFRVP protocol. Given two plain texts m_1, m_2 with their respective encryptions $E(m_1)$, $E(m_2)$, the multiplicative homomorphic property (possessed by the ElGamal and partially by the BGN ciphers) states that $E(m_1) \odot E(m_2) = E(m_1 \cdot m_2)$, where $\odot$ is an arithmetic operation in the encrypted domain that is equivalent to the usual multiplication operation in the plain text domain. The additive homomorphic property (possessed by the BGN and the Paillier schemes) states that $E(m_1) \oplus E(m_2) = E(m_1 + m_2)$, where $\oplus$ is an arithmetic operation in the encrypted domain which is equivalent to the usual sum operation in the plain text domain.

C. Distance Computations

As discussed earlier, the fair rendez-vous point L_{fair} is the location preference that minimizes the maximum distance between any other location preference and L_{fair} . In our algorithms, we minimize with respect to the *square* of the distances, because distance squares are much easier to compute in an oblivious fashion (by using homomorphic encryptions) than simple distances. As the squaring function is order preserving, the problem of finding the argument that minimizes the maximum distance is equivalent to finding the argument that minimizes the maximum *squared* distance. *BGN-Distance:* Our first distance computation algorithm is based on the BGN encryption scheme. This novel protocol requires only one round of communication between each user and the LDS, and it efficiently uses both the multiplicative and additive homomorphic properties of the BGN scheme.

D. The **PPFRVP** *Protocol*

The PPFRVP protocol has three main modules: (A) the distance computation module, (B) the MAX module and (C) the ARGMIN MAX module.

1) Distance Computation: The distance computation module uses either the BGN-distance or the Paillier-ElGamaldistance protocols. We note that modules (B) and (C) use the same encryption scheme as the one used in module (A).

2) MAX Computation: In Step B.1, the LDS needs to hide the values within the encrypted elements before sending them to the users. This is done to avoid disclosing private information, such as the pairwise distances or location preferences, to users.3 In order to mask these values, for each index *i*, the LDS generates two random values (*ri* and *si*) that are used to scale and shift the *ctot i j* (the encrypted square distance between *Li* and *L j*) for all *j* , thus, obtaining d^*_{ij} . This is done in order to (i) ensure privacy of real pairwise distances, (ii) be resilient in case of collusion among users and (iii) preserve the internal order (the inequalities) among the pairwise distance from each user to all other users. Afterwards, in Step B.2 the LDS chooses two private element-permutation functions σ (for *i*) and θ (for *j*) and permutes d^*_{ij} , obtaining the permuted values $d^*_{\sigma i}\ \theta\ j$, where $i, j \in \{1, . . . , N\}$. The LDS sends *N* such distinct elements to each user. In Step B.3, each user decrypts the received values, determines their maximum and sends the index *σmax i* of the maximum value to the LDS. In Step B.4 of the MAX module (B), the LDS inverts the permutation functions σ, θ and removes the masking from the received indexes corresponding to the maximum distance values.

3) ARGMIN MAX Computation: In Step C1, the LDS masks the true maximum distances by scaling and shifting them by the same random amount such that their order is preserved. Then, the LDS sends to each user all the masked maximum distances. In Step C2, each user decrypts the received masked (randomly scaled and shifted) maximum values, and determines the minimum among all maxima. In Step C3, each user knows which identifier corresponds to himself, and the user whose preferred location has the minimum distance sends to all other users the fair rendezvous location in an anonymous way. After the last step, each user receives the final FRV location, but no other information regarding non-fair locations or distances is leaked.

V. Privacy Requirements And Definitions

Informally, the privacy requirements can be stated as follows. After the execution of the PPFRVP algorithm, any participating user u_i should not be able to infer (i) the preferred location L_j of any other user u_j , $u_j \neq u_i$ nor (ii) the relative distances d_{ij} between any two users u_i and u_j . Likewise, any LDS should not be able to infer (iii) the preferred location L_i of any user u_i , (iv) the relative distance d_{ij} between any two users u_i and u_j nor (v) the final rendez-vous location L_{fair} . Formally, these privacy requirements can be classified as *user privacy* and *server privacy*, as defined below.

A. User Privacy

The *user-privacy* of any PPFRVP algorithm *A* measures the probabilistic advantage that an adversary u_a gains towards learning the preferred location of at least one other user, except the final fair rendez-vous location, after all users have participated in the execution of the PPFRVP protocol. An adversary in this case is a user participating in *A*. We express user-privacy as three different probabilistic advantages. First, we measure the probabilistic advantage of an adversary u_a in correctly guessing the preferred location L_i of any user $u_i \neq u_a$. This is referred to as the *identifiability advantage* and is denoted by $Adv_a^{IDT}(A)$. We will define $Adv_a^{IDT}(A)$ using a challenge-response game. Let $\mathbb{U} = \{u_1, . . . , u_N , u_a\}$ be the set of all users, including the adversary u_a, C be the challenger, and *A* be the proposed PPFRVP algorithm. Then, the identifiability game is defined as follows:

where *Pr(k' = k)* is the probability that u_a correctly guesses the value *k* chosen by the challenger. The above notion of identifiability is also called *weak identifiability* because the adversary knows that the challenge belongs to one of the participant. A stronger notion of identifiability can also be defined, where the challenge is a randomly chosen two-dimensional position coordinate not necessarily belonging to one of the participating users. The adversary in this game wins if he correctly guesses if the challenge location belongs to one of the participants or not. In this work, we focus only on the weak identifiability property.

The second measure of user-privacy is the *distancelinkability advantage*, which is the probabilistic advantage of an adversary u_a in correctly guessing whether the distance d_{ij} between any two participating users $u_i \neq u_j$, is greater than a given parameter *s*, without learning any users' preferred locations L_i , L_j .

B. Server Privacy

For the third-party (LDS) adversary, the game definitions are similar to those defined for an user adversary, except that the LDS does not receive L_{fair} in the Step 2 of the game. Then, the server-privacy of a PPFRVP algorithm *A* can then be defined as follows. In practice, users will execute the PPFRVP protocol multiple times with either similar or completely different sets of participating users, and with the same or a different location preference in each execution instant.

C. Overall PPFRVP Privacy

Based on the above definitions of user and server-privacy, we are now ready to express the overall privacy requirements of any PPFRVP algorithm. Before that, let us first define a *private* execution.

Definition 4: A private *execution of any PPFRVP algorithm A is an execution which does not reveal more information than what can be derived from the inputs, the intermediate results and its output.*

Based on how memory is retained over sequential executions, we define two types of algorithm executions, namely, *dependent* and *independent.*

Definition 5: An independent *(respectively,* dependent*)* execution *is a single private execution of the PPFRVP algorithm in which* no *(respectively,* some*) information of an earlier and current execution is retained and passed to future executions.* The information that might be transferred from an earlier execution to the next can include past inputs, intermediate results and the outputs of the algorithm. Based on the type of execution, the privacy conditions of a privacy-preserving meeting-location algorithm can be defined as follows.

Definition 6: A PPFRVP algorithm A is execution *(respectively,* fully*) privacy-preserving if and only if for every* independent *(respectively,* all*) execution(s)*

1) *A is correct; All users are* correctly *able to compute the final fair rendez-vous location* L_{fair} *;*

2) *A is* user-private*;*

3) *A is* server-private.

A fully privacy-preserving PPFRVP algorithm is a much stronger (and difficult to achieve) privacy requirement. Initially, we focus on analyzing the independent execution privacy of our proposed PPFRVP algorithm. Later, we briefly analyze privacy-leakage due to dependent executions.

VI. Experimental Evaluation

In this section, we present an in-depth evaluation of the proposed PPFRVP protocols by outlining the results of controlled experiments and user studies conducted using prototype implementation of the protocols on modern mobile devices.

A. Implementation and Performance Measurements

The BGN-based PPFRVP protocol, we measure the performance using both a 160-bit and a 256-bit secret key, whereas for the ElGamal-Paillier-based protocol we use 1024-bit secret keys. A 160-bit key in elliptic curve-based cryptosystems such as BGN provides equivalent security as a 1024-bit key in RSA and ElGamal cryptosystems.

1) Computation Delay on the LDS: We can computation time required by the LDS increases with the number of users. Moreover, the ElGamal-Paillier based scheme is the most efficient across all computations, requiring only 4 seconds executing the protocol with 10 participants. The two BGN-based algorithms are less efficient execution-wise (9 seconds). This is due to the CPU-intensive bilinear mapping operations of the BGN cryptosystem. For the modules B and C, the BGN-based algorithms outperform the one based on ElGamal-Paillier. The maximum computations on the LDS require 0.5 seconds for the 160-bit BGN algorithm, whereas the ElGamal-Paillier takes almost 2 seconds. A similar result can be observed for the minimum function computations. There are two main reasons for this. First, there are no bilinear mappings involved in these modules and second, the BGN-based algorithms use much smaller key sizes. From a practical perspective, both the ElGamal-Paillier and the BGN algorithms have good performance in modules B and C of the PPFRVP protocol.

2) Computation Delay on the Nokia N810 Clients: The different computation times on the Nokia N810 mobile device. As it can be seen, our BGN-based algorithm is the most efficient for the distance computations, requiring only 0.3 seconds, independently of the number of users. This is possible because each client needs to send only once its own encrypted vectors in order to allow the LDS to compute all pairwise distances, as opposed to the ElGamal- Paillier based algorithm that requires users to decrypt and re-encrypt values multiple times (depending on the number of users). The alternative protocol, on the contrary, needs 4 seconds with 10 participants. However, in the subsequent phases, the results are not as good because the BGN-based protocol makes intensive use of bilinear mapping operations. Overall,

3) Power Consumption Analysis on the Nokia N810 Clients: In order to analytically evaluate the power consumption of the PPFRVP protocol computations, the authors propose a fairly accurate non-linear model for measuring power consumption of Nokia N810 devices, which uses parameters that can be obtained easily by the operating system at runtime.

B. User Study

In order to assess users' privacy-related preferences in LSB services and to get an opinion on the usability of our prototypes, we conduct a targeted user-study on 35 respondents, sampling a population of technology-savvy college students (in the age group of 20–30 years) and non-scientific personnel.

User-Study: The user study consists of three phases: Phase 1 - to assess the participants' level of adoption of mobile LSBS and their sensitivity to privacy issues, respondents answered a set of 22 general questions on LSBS and privacy concerns. The answers to these questions are either "Yes" or "No", or on a 4-point Lickert scale (where 1 means *Disagree*, 4 is *Agree*). 2) Phase 2 - respondents were instructed to use our prototype mobile FRVP application. 3) Phase 3 - after using the prototype application, participants answer a set of 12 questions by choosing an answer from a 4-point Lickert scale. The purpose of phase 3 is to evaluate the usability of our application, and to assess whether privacy undermines its usability or performance from the end-user's perspective.

Phase 1: A majority of the participants in our user study are males in the age-group of 20–25 years. Around 86% of them use social networks, and 74% browse the Internet with a mobile device. When organizing meetings, 54% of the time they involve groups of 4 people and 29% groups of 6 individuals, and participants use their mobile device for organizing 63% of such meetings. Although only 14% are aware of existing location sharing-based applications, 51% would be *very* or *quite* interested in using an application such as the FRVP. However, they are sensitive to privacy (98%) and anonymity (74%) in their online interactions, especially with respect to the potential misuse of their private information by non-specified third-parties (88%). Most of the participants (80%) agree that their personal information should not be disseminated without their knowledge.

These results indicate that LSBSs are perceived as interesting by the majority of the sampled population, who are also the most likely to adopt LBS technologies [4]. They also agree that privacy is crucial for the acceptability of such services.

VII. Extension To Multiple Preferences

The PPFRVP protocol, allows each user *i* to select one preferred location L_i in order to determine the fair rendez-vous location L_{fair}. A natural extension of the existing protocol is to allow any user *i* to select multiple preferred locations $L_{i,1}, \ldots, L_{i},v_{i}$. In this way, the users would have more flexibility in choosing location preferences. Moreover, users could assign a priority or weight to each location in their set of preferences. The PPFRVP protocol with multiple location preferences. Multiple preferences are included in the PPFRVP protocol by assigning a priority to each preferred location *Li, j* for all users *i* and preferences *j* . One way to include them in the distance computations is to assign weighting coefficients *pi, j* for the maximum distances *c(Li, j , Lk,h)* computed at the end of Step 3; this way, the highest priority could be defined by using the lowest value of *pi*,1 = 1, whereas the lower priorities could be assigned higher values of *pi*,2 = *pi*,3 = 2. As a result, the minimum of these maximum metadistance is crucial for each client in order to select his own prioritized location in Step 3.1, which will then be sent to the LDS for the continuation of the PPFRVP computations.

VIII. Related Work

The problem of privacy-preserving fair rendez-vous location has received little or no attention in the literature. The survey of existing literature on meeting-location algorithms and propose a more comprehensive solution for such a problem. Although considering aspects such as user preferences and constraints, their work (or the surveyed papers) does not address any security or privacy issues. Similarly, an efficient meeting-location algorithm that considers the time in-between two consecutive meetings. However, all private information about users is public. In the domain of Secure Multiparty Computation (SMC), several authors have addressed privacy issues related to the computation of the distance between two routes or points. According to SMC protocols for securely computing the distance between a point and a line segment, the distance between two moving points and the distance between two line segments design and implement three distributed privacy-preserving protocols for nearby friend discovery, and they show how to cryptographically compute the distance between a pair of users. However, due to the fully distributed nature of the aforementioned approaches, the computational and communication complexities increase significantly with the size of the participants and inputs. Moreover, all parties involved in the computations need to be online and synchronized. There have also been several research results in the literature that focus on the problem of privacy-preserving *location-based queries* and *location sharing* or anonymous *location checkins*. However, these research efforts attempt to solve issues that are orthogonal, and uniquely different, from the ones addressed in this paper. In the direction of anonymous location sharing, a novel architecture called *ZeroSquare* where the main goal is to provide a location hub for privacy-preserving geospatial applications. The main idea of the authors is to decouple user (profile) information from location information by assuming two non-colluding entities that store this information. Their work is different from ours in that they do not consider the problem of privately computing some function based on the location data, rather they want to enable privacy-preserving location sharing in mobile applications. Contrary to the work, a privacy-preserving system that allows users to set location-triggered alarms based on presence at specific locations, rather than sharing location coordinates and also propose a set of privacy-preserving protocols, using well-known cryptographic constructs, which anonymously proves to a venue that a user checked-in a certain number of times.

IX. Conclusion And Future Work

In this work, we addressed the privacy issue in the Fair Rendez-Vous Problem (FRVP). Our solutions are based on the homomorphic properties of well-known cryptosystems. We designed, implemented and evaluated the performance of our algorithms on real mobile devices. We showed that our solutions preserve user preference privacy and have acceptable performance in a real implementation. Moreover, we extended the proposed algorithms to include cases where users have several prioritized locations preferences. Finally, based on an extensive user-study, we showed that the proposed privacy features are crucial for the adoption of any location sharing or location-based applications.

References

[1] (2011, Nov.). Facebook Statistics [Online]. Available: http://www.facebook.com/ /press / info.php? statistics

[2] (2011, Nov.). Facebook Deals [Online]. Available: http://www.facebook.com/deals/

[3] E. Valavanis, C. Ververidis, M. Vazirgianis, G. C. Polyzos, and K. Norvag, “MobiShare: Sharing context-dependent data & services from mobile sources,” in Proc. IEEE/WIC Int. Conf. WI, Oct. 2003, pp. 263–270.

[4] (2011). Microsoft Survey on LBS [Online]. Available: http://go.microsoft.com/?linkid=9758039

[5] (2011, Nov.). Orange Taxi Sharing App [Online]. Available: http://event.orange.com/default/EN/all/mondial

[6] (2011). Let’s Meet There [Online]. Available:http://www.letsmeetthere.net/

[7] P. Golle and K. Partridge, “On the anonymity of home/work location pairs,” in Proc. 7th Int. Conf. Pervasive Computing, 2009, pp. 390–397.

[8] . Freudiger, R. Shokri, and J.-P. Hubaux, “Evaluating the privacy risk of location-based services,” in Proc. 15th Int. Conf. Financial, 2011, pp. 31–46.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019. Journal no. 49102.