Privacy-Preserving Optimal Meeting Location Determination on Mobile Devices

Igor Bilogrevic, *Member, IEEE*, Murtuza Jadliwala, *Member, IEEE*, Vishal Joneja, Kübra Kalkan, Jean-Pierre Hubaux, *Fellow, IEEE*, and Imad Aad

***Abstract*— Equipped with state-of-the-art smartphones and mobile devices, today's highly interconnected urban population is increasingly dependent on these gadgets to organize and plan their daily lives. These applications often rely on current (or preferred) locations of individual users or a group of users to provide the desired service, which jeopardizes their privacy; users do not necessarily want to reveal their current (or preferred) locations to the service provider or to other, possibly untrusted, users. In this paper, we propose privacy-preserving algorithms for determining an optimal meeting location for a group of users. We perform a thorough privacy evaluation by formally quantifying privacy-loss of the proposed approaches. In order to study the performance of our algorithms in a real deployment, we implement and test their execution efficiency on Nokia smartphones. By means of a targeted user-study, we attempt to get an insight into the privacy-awareness of users in location-based services and the usability of the proposed solutions.**

***Index Terms*— Mobile application, oblivious computation, privacy.**

I. Introduction

THE rapid proliferation of smartphone technology in urban communities has enabled mobile users to utilize context-aware services on their devices. Service providers take advantage of this dynamic and ever-growing technology landscape by proposing innovative context-dependent services for mobile subscribers. Location-based Services (LBS), for example, are used by millions of mobile subscribers every day to obtain location-specific information [1].

Two popular features of location-based services are *location check-ins* and *location sharing*. By checking into a location, users can share their current location with family and friends or obtain location-specific services from third-party providers [2], [3]. The obtained service does not depend on the locations of other users. The other type of location-based services, which rely on sharing of locations (or location preferences) by a group of users in order to obtain some service for the whole group, are also becoming popular. According to a recent study [4], location sharing services are used by almost 20% of all mobile phone users. One prominent example of such a service is the taxi-sharing application, offered by a global telecom operator [5], where smartphone users can share a taxi with other users at a suitable location by revealing their departure and destination locations. Similarly, another popular service [6] enables a group of users to find the most geographically convenient place to meet.

Privacy of a user's location or location preferences, with respect to other users and the third-party service provider, is a critical concern in such location-sharing-based applications. For instance, such information can be used to de-anonymize users and their availabilities [7], to track their preferences [8] or to identify their social networks [9]. For example, in the taxi-sharing application, a curious third-party service provider could easily deduce home/work location pairs of users who regularly use their service. Without effective protection, even sparse location information has been shown to provide reliable information about a users' private sphere, which could have severe consequences on the users' social, financial and private life [10], [11]. Even service providers who legitimately track users' location information in order to improve the offered service can inadvertently harm users' privacy, if the collected data is leaked in an unauthorized fashion or improperly shared with corporate partners. Recent user studies [4] show that end-users are extremely sensitive about sharing their location information. Our study on 35 participants, including students and non-scientific staff, showed that nearly 88% of users were not comfortable sharing their location information. Thus, the disclosure of private location in any Location-Sharing-Based Service (LSBS) is a major concern and must be addressed.

In this paper, we address the privacy issue in LSBSs by focusing on a specific problem called the *Fair Rendez-Vous Point (FRVP)* problem. Given a set of user location preferences, the FRVP problem is to determine a location among the proposed ones such that the maximum distance between this location and all other users' locations is minimized, i.e. it is *fair* to all users. Our goal is to provide practical privacy-preserving techniques to solve the FRVP problem, such that neither a third-party, nor participating users, can learn other users' locations; participating users only learn the optimal location. The privacy issue in the FRVP problem is representative of the relevant privacy threats in LSBSs.

Manuscript received October 3, 2013; revised December 30, 2013 and March 1, 2014; accepted April 7, 2014. Date of publication April 18, 2014; date of current version June 17, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Jianying Zhou. *(Igor Bilogrevic and Murtuza Jadliwala contributed equally to this work.)*

I. Bilogrevic and J.-P. Hubaux are with the Swiss Federal Institute of Technology, Lausanne 1015, Switzerland (e-mail: igor.bilogrevic@gmail.com; jean-pierre.hubaux@epfl.ch).

M. Jadliwala is with Wichita State University, Wichita, KS 67260 USA (e-mail: murtuza.jadliwala@wichita.edu).

V. Joneja is with ELCA Informatique, Lausanne 1007, Switzerland (e-mail: vishal.joneja@epfl.ch).

K. Kalkan is with Sabanci University, Istanbul 34956, Turkey (e-mail: kubrakalkan@sabanciuniv.edu).

I. Aad is with the University of Bern, Bern 3012, Switzerland (e-mail: aad@iam.unibe.ch).

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

Digital Object Identifier 10.1109/TIFS.2014.2318435

Our contributions in this paper are as follows. We first formulate the FRVP problem as an optimization problem, specifically the *k-center* problem [12], and then analytically outline the privacy requirements of the participants with respect to each other and with respect to the solver (in this case, a third-party service provider). We then propose two algorithms for solving the above formulation of the FRVP problem in a privacy-preserving fashion, where each user participates by providing only a single location preference to the FRVP solver or the service provider. Our proposed algorithms take advantage of the homomorphic properties of well-known cryptosystems, such as BGN, ElGamal and Paillier, in order to privately compute an optimally fair rendez-vous point from a set of user location preferences. In this significantly extended version of our earlier conference paper [13], we evaluate the security of our proposal under various passive and active adversarial scenarios, including collusion. We also provide an accurate and detailed analysis of the privacy properties of our proposal and show that our algorithms do not provide any probabilistic advantage to a passive adversary in correctly guessing the preferred location of any participant. In addition to the theoretical analysis, we also evaluate the practical efficiency and performance of the proposed algorithms by means of a prototype implementation on a testbed of Nokia mobile devices. We also address the multi-preference case, where each user may have multiple prioritized location preferences. We highlight the main differences, in terms of privacy and performance, with the single preference case, and also present initial experimental results for the multi-preference implementation. Finally, by means of a targeted user study, we provide insight into the usability of our proposed solutions.

II. System Architecture

We consider a system composed of two main entities: (i) a set of users[1] (or mobile devices) $\mathbb{U} = \{u_1, \ldots, u_N\}$ 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 user-preferred 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 u_i has the means to determine the coordinates $L_i = (x_i, y_i) \in \mathbb{N}^2$ of his preferred rendez-vous location. We consider a two-dimensional coordinate system, but the proposed schemes are general enough and can be easily extended 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.[2]

[1] *users*, *participants* and *devices* are used interchangeably.

[2] http://www.gps.gov/systems/gps/performance/accuracy/

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 rendez-vous 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 [16] 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 $\mathbb{L} = \{L_i\}_{i=1}^{N}$. 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 $(K_P^{u_i}, K_s^{u_i})$, 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 $(K_P^{M_v}, K_s^{M_v})$ in an online fashion for each meeting setup instance v. The private key $K_s^{M_v}$ generated in this way is known only to all meeting participants, whereas the public key $K_P^{M_v}$ is known to everyone including the LDS. This could be achieved by means of a secure credential establishment protocol [17], [18].

The LDS executes the FRVP algorithm on the inputs it receives from the users in order to compute the FRV point. The LDS is also able to perform public-key cryptographic functions. For instance, a common public-key infrastructure using the RSA cryptosystem [19] 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 [20] 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 (as opposed to [21]). 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: Similar to the LDS assumption, 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 FRVP 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 rendez-vous point among a set of user-proposed locations, such that (i) the rendez-vous point is *fair* (as defined in Section IV-A) 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.

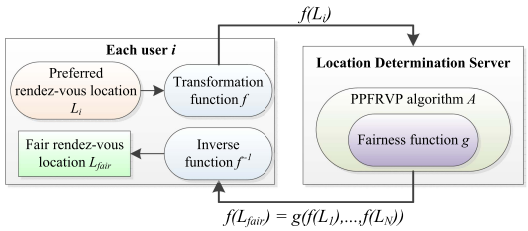


Fig. 1. Functional diagram of the PPFRVP protocol.

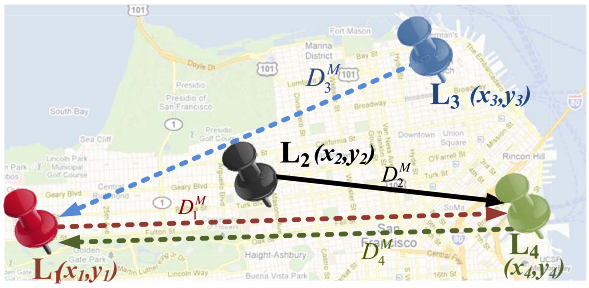


Fig. 2. PPFRVP scenario, where the fairness function is $g = \text{argmin}_i(D_i^M)$. The dashed arrows represent the maximum distance D_i^M from each user u_i to any user $j \neq i$, whereas the solid line is the minimum of all such maximum distances. The fair rendez-vous location is $L_{fair} = L_2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

IV. Proposed Solution to PPFRVP Problem

In this section, we outline the details of our proposed protocol for solving the PPFRVP 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 $\mathbb{U} = \{u_i\}_{i=1}^N$ 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 $(L_1, \ldots, L_k)$ 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 rendez-vous 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 $L_1, \ldots, L_N$ 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, as outlined in Section VIII. Let $d_{ij} \geq 0$ be the Euclidean distance between two points $L_i, L_j \in \mathbb{N}^2$, and $D_i^M = \max_{j \neq i} d_{ij}$ 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* $L_{fair} \in \mathbb{L} = \{L_1, \ldots, L_N\}$, *where fair* $= arg\min_i D_i^M$.

Thus, a solution to the PPFRVP problem privately (w.r.t. the LDS and the participating users) determines the fair rendezvous location as that user-proposed location preference which is closest to all other proposed locations, as compared to any other proposed location prefernces. In order for the LDS to *privately* compute the fair rendez-vous location, the fairness function g would be required to operate in an oblivious fashion, i.e., without having access to the location preferences L_i. This can be accomplished using cryptographic schemes with homomorphic encryption properties, as discussed next.

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, specifically, *Boneh-Goh-Nissim* (BGN) [22], *ElGamal* [23] and *Paillier* [24] cryptosystems, 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. Further details of these cryptosystems (and their homomorphic properties) can be found in [22], [23], and [24].

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.

1) 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. The BGN-distance protocol works as follows. First, each user u_i, $\forall i \in \{1, \ldots, N\}$, creates the vectors

$$\begin{aligned} E_i(a) =< a_{i1} | \ldots | a_{i6} >=< & E(x_i^2) | E(T - 2x_i) | E(1) \\ & | E(T - 2y_i) | E(y_i^2) | E(1) > \\ E_i(b) =< b_{i1} | \ldots | b_{i6} >=< & E(1) | E(x_i) | E(x_i^2) \\ & | E(y_i) | E(1) | E(y_i^2) > \end{aligned}$$

where, $E(\cdot)$ is the encryption using the BGN scheme with the fresh session key $K_P^{M_v}$, $L_i = (x_i, y_i)$ is the desired rendez-vous location of user u_i and T is the modulus of the plaintext domain. Afterwards, each user sends the two vectors $E_i(a)$, $E_i(b)$ over a secure channel to the LDS. Then, the LDS computes the scalar product $E_i(a) \cdot E_j(b)$ of the received vectors, which produces the encrypted pairwise distances $E(d_{ij}^2)$ by first applying the multiplicative and then the additive homomorphic property of BGN. For example, in a scenario with two users, one can easily verify that

$$\begin{aligned} E_i(a) \bullet E_j(b) &= E(x_i^2 + x_j(T - 2x_i) + x_j^2 \\ &\quad + y_j(T - 2y_i) + y_i^2 + y_j^2 \mod T) \\ &= E(d_{ij}^2 \mod T) \end{aligned}$$

where T is chosen such that $\forall i, j \in \{1, \ldots, N\}, d_{ij}^2 < T$. At this point, the LDS has obliviously computed $E(d_{ij}^2)$, which is the (encrypted) square of the pairwise distances between all pairs L_i, L_j of user-desired locations, where $i \neq j$.

2) Paillier-ElGamal-Distance: An alternative scheme for the distance computation is based on both the Paillier and ElGamal encryptions, as shown in Fig. 3. In addition to the multiplicative homomorphic property of ElGamal, we rely on the two following properties of the Paillier encryption:

$$E(m_1) \cdot E(m_2) = E(m_1 + m_2 \mod n), \quad \forall m_i \in \mathbb{Z}_n \quad (1)$$

$$E(m_1)^r = E(r \cdot m_1 \mod n), \quad \forall r \in \mathbb{Z}_n^* \quad (2)$$

which implies that

$$\begin{aligned} E(r \cdot m_1)^{r^{-1}} &= E(r^{-1} \cdot r \cdot m_1 \mod n) \\ &= E(m_1 \mod n) \end{aligned} \quad (3)$$

where r^{-1} is the multiplicative inverse of $r \mod n$. As neither Paillier or ElGamal possess both multiplicative and additive properties, the resulting algorithm requires one extra step in order to obliviously compute the pairwise squared distances d_{ij}^2 [13]. We initialize the ElGamal scheme in a suited algebraic group $\mathbb{Z}_q$ and the Paillier in $\mathbb{Z}_n$, where q is a large prime and independent from $n = pz$, with p, z large primes as well, such that $\gcd(pz, (p-1)(z-1)) = 1$. To provide an equivalent level of security, we assume that $|q| = |n|$, i.e., the

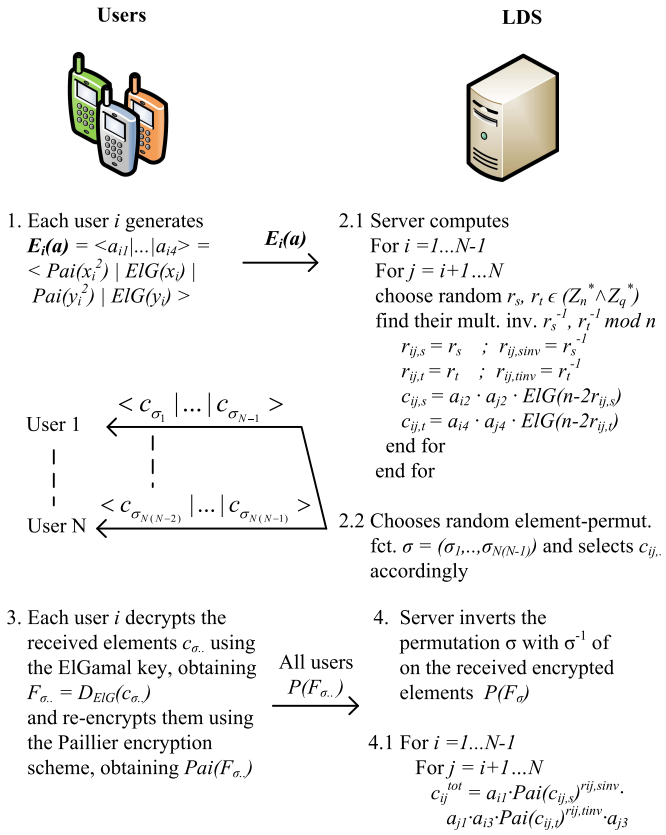


Fig. 3. Privacy-preserving distance computation protocol based on the ElGamal and Paillier encryption schemes.

size of the moduli is the same (e.g., 1024 bits). Moreover, in this scheme the participating users derive two pairs of public/private session keys $\{(K_P^{Mv1}, K_s^{Mv1}), (K_P^{Mv2}, K_s^{Mv2})\}$ from the shared secret, where the pair $v1$ is used with the ElGamal encryption scheme and $v2$ with the Paillier one. The public keys, as well as the public parameters including n and q, are known to all users and to the LDS. To simplify the notation, in the following we omit the indices of the cryptographic keys, as they are clear from the context.

The distances are computed as follows. First, each user u_i, $\forall i \in \{1, \ldots, N\}$, creates the vector

$$\begin{aligned} E_i(a) &= < a_{i1} | \ldots | a_{i4} > \\ &= < Pai(x_i^2) | ElG(x_i) | Pai(y_i^2) | ElG(y_i) > \end{aligned}$$

Afterwards, each user u_i sends the vector $E_i(a)$ to the LDS, encrypted with LDS's public key. In Step 2.1, the LDS computes the scalar product of the second and fourth element of the received vectors (as shown in Fig. 3). In order to hide this intermediate result from the users, the LDS obliviously randomizes these results with random values $r_s, r_t \in (\mathbb{Z}_q^* \wedge \mathbb{Z}_n^*)$, such that r_s, r_t are relative primes. This ensures that r_s, r_t have multiplicative inverses in the Paillier group modulo n. Moreover, to guarantee that the randomization and de-randomization in the following steps will generate a coherent result, it is required that $0 < x_i, x_j(n - 2r_s) < n$, for any possible value of x_i, x_j (and respectively for r_t and y_i, y_j). Therefore, the LDS selects r_s, r_t such that:

$$\frac{n}{2} - \frac{n}{2x_i x_j} < r_s < \frac{n}{2}, \qquad \frac{n}{2} - \frac{n}{2y_i y_j} < r_t < \frac{n}{2} \tag{4}$$

Hence, there are still $\frac{n}{2(\max_i x_i)^2}$ possible different values for the choice of r_s (and similarly for r_t). In practice, assuming that $|n| = 1024$ bits (where $2^{1024} \approx 10^{309}$) and that the coordinates (x_i, y_i) are expressed in meters ($0 < \max_i x_i, \max_i y_i < 10^8$), the LDS could still choose random values r_s, r_t among a total of $5 \cdot 10^{292} > 2^{970}$ possible values.

After choosing r_s, r_t, the LDS computes their inverses, denoted as r_s^{-1}, r_t^{-1}. The randomized scalar products are denoted as $c_{ij,s}$ and $c_{ij,t}$. In Step 2.2, the LDS permutes the order of all $c_{ij,s}$ and $c_{ij,t}$ with its private element-permutation function $\sigma = [\sigma_1, \ldots, \sigma_{N(N-1)}]$, and sends N such distinct elements to each user u_i. In Step 3, each user simply decrypts the received elements with the ElGamal private key K_s^{Mv1} and re-encrypts them with the Paillier public key K_P^{Mv2}. Then, each user sends the re-encrypted elements to the LDS in the same order as he received it. In Step 4, the LDS reverts the element-permutation function σ, and in Step 4.1 it finally computes the d_{ij}^2 for all i, j, after having removed the randomizing factors $r_{ij,s}, r_{ij,t}$ with their inverses $r_{ij,sinv}$ and $r_{ij,tinv}$ as shown in Eqn. (3). At this point, the LDS has securely computed $E(d_{ij}^2)$, the (encrypted) square of the pairwise distances between all pairs of user-desired locations $L_i \neq L_j$.

D. The PPFRVP *Protocol*

The PPFRVP protocol (shown in Fig. 4) 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-ElGamal-distance protocols. We note that modules (B) and (C) use the same encryption scheme as the one used in module (A). In other words, $E(.)$ in Fig. 4 refers to encryption using either the BGN or the Paillier encryption scheme.

2) MAX Computation: In Step B.1, the LDS needs to hide the values within the encrypted elements (i.e., the pairwise distances computed earlier) 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 (r_i and s_i) that are used to scale and shift the c_{ij}^{tot} (the encrypted square distance between L_i 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 $i, j \in \{1, \ldots, N\}$. 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

[3]After the distance computation module (A), the LDS possesses all encrypted pairwise distances. This encryption is made with the public key of the participants and thus the LDS cannot decrypt the distances without the corresponding private key. The oblivious (and order-preserving) masking performed by the LDS at Step B.1 is used in order to hide the pairwise distances from the users themselves, as otherwise they would be able to obtain these distances and violate the privacy of the users.

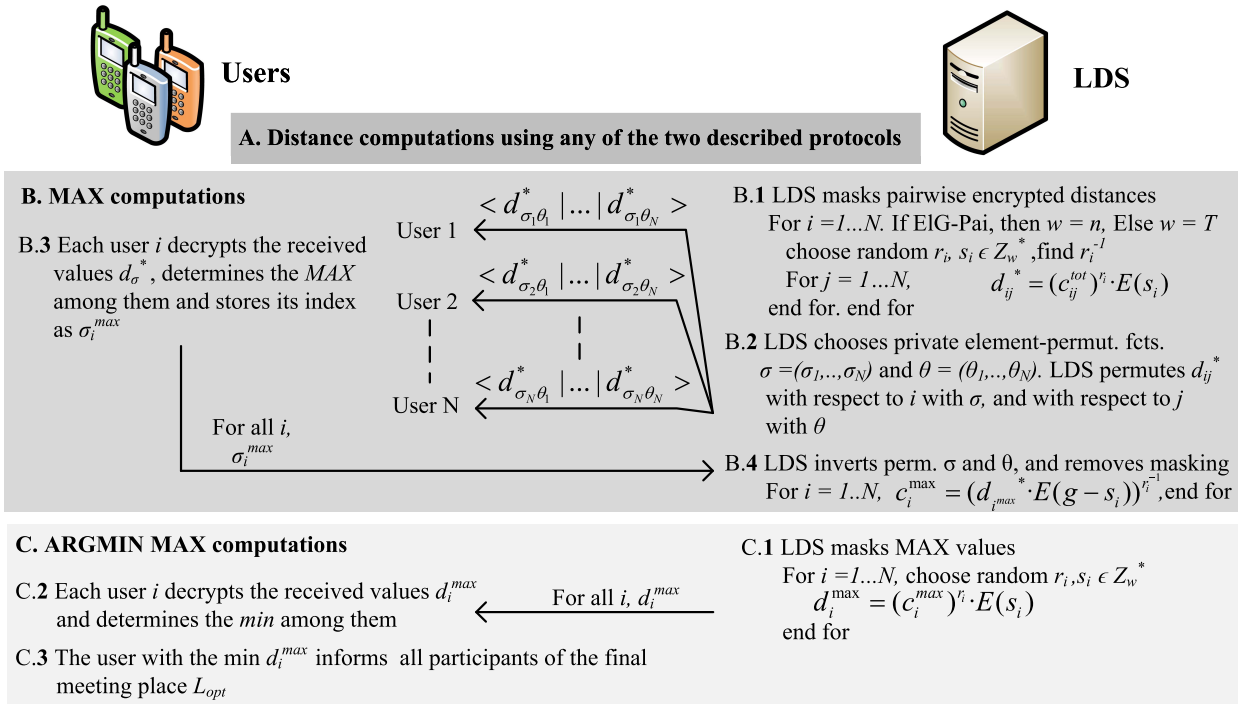


Fig. 4. Privacy-Preserving Fair Rendez-Vous Point (PPFRVP) protocol. In Phase A, the LDS obliviously computes all squared pairwise distances. In Phase B, the LDS computes the maxima of the pairwise distances with by involving the users. In Phase C, the users determine the fair rendez-vous location L_{fair}.

index σ_i^{max} 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 C.1, 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 C.2, each user decrypts the received masked (randomly scaled and shifted) maximum values, and determines the minimum among all maxima. In Step C.3, 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 rendez-vous location in an anonymous way. After the last step, each user receives the final fair rendez-vous 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 $\mathbb{U} = \{u_1, \ldots, u_N, u_a\}$ be the set of all users, including the adversary u_a, $\mathbb{C}$ be the challenger, and A be the proposed PPFRVP algorithm. Then, the identifiability game is defined as follows:

1) Challenger setup: $\mathbb{C}$ privately collects the preferred rendez-vous locations L_i; $L_i \neq L_j, \forall i, j \in \{1, \ldots, N\}$.
2) Algorithm execution: $\mathbb{C}$ executes the PPFRVP algorithm A with all users $\mathbb{U}$ and computes $f(L_{fair}) = g(f(L_1), \ldots, f(L_N), f(L_a))$. It then sends $f(L_{fair})$ to each user $u_i \in \mathbb{U}$.
3) Challenge: $\mathbb{C}$ chooses a random $k \in \{1, \ldots, N\}$ and sends L_k to the adversary u_a.
4) Guess: u_a chooses a value $k' \in \{1, \ldots, N\}$ and sends it back to the challenger. u_a wins the game if $k' = k$, otherwise he loses.

Then, the identifiability advantage $Adv_a^{IDT}(A)$ is the probabilistic advantage of the adversary in winning this game:

$$\begin{aligned} Adv_a^{IDT}(A) &= Pr[u_a \text{ wins the game}] - 1/N \\ &= Pr[k' = k] - 1/N \end{aligned} \tag{5}$$

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 (in Step 3) 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 *distance-linkability 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. We denote this advantage as Adv_a^{d-LNK}. The distance-linkability game is defined as follows.

1) Challenger setup: $\mathbb{C}$ privately collects the preferred rendez-vous locations $L_i \neq L_j, \forall i, j \in \{1, \ldots, N\}$.
2) Algorithm execution: $\mathbb{C}$ executes the PPFRVP algorithm A with all users $\mathbb{U}$ and computes $f(L_{fair}) = g(f(L_1), \ldots, f(L_N), f(L_a))$. It then sends $f(L_{fair})$ to each user $u_i \in \mathbb{U}$.
3) Challenge: $\mathbb{C}$ chooses a random value s and two distinct users $u_j, u_k, \forall j, k \in \{1, \ldots, N\}, j \neq k$. $\mathbb{C}$ sends (j, k, s) to the adversary.
4) Guess: u_a responds with a value $s^* \in \{0, 1\}$. u_a wins the game if $s^* = 0$ and $d_{j,k} \geq s$, or if $s^* = 1$ and $d_{j,k} < s$. Otherwise, the adversary looses.

The distance-linkability advantage $Adv_a^{d-LNK}(A)$ is the probabilistic advantage of the adversary in winning this game:

$$Adv_a^{d-LNK}(A) = Pr[s^* = 0 \wedge d_{j,k} \geq 0] + Pr[s^* = 1 \wedge d_{j,k} < 0] - 1/2 \quad (6)$$

Lastly, the *coordinate-linkability advantage*, denoted as Adv_a^{c-LNK}, is the probabilistic advantage of an adversary u_a in correctly guessing whether a given coordinate x_i (or y_i) of a user u_i is greater than the corresponding coordinate(s) of another user $u_j \neq u_i$ without learning the users' preferred locations L_i, L_j. The coordinate-linkability game is as follows.

1) Challenger setup: $\mathbb{C}$ privately collects the preferred rendez-vous locations $L_i \neq L_j, \forall i, j \in \{1, \ldots, N\}$.
2) Algorithm execution: $\mathbb{C}$ executes the PPFRVP algorithm A with all users $\mathbb{U}$ and computes $f(L_{fair}) = g(f(L_1), \ldots, f(L_N), f(L_a))$. It then sends $f(L_{fair})$ to each user $u_i \in \mathbb{U}$.
3) Challenge: $\mathbb{C}$ throws an unbiased coin to select a coordinate axis $b| \in \{x, y\}$. $\mathbb{C}$ randomly chooses $j, k \in \{1, 2, \ldots, N\}, j \neq k$. $\mathbb{C}$ sends $\{j, k, b\}$ to u_a as a challenge.
4) Guess: u_a responds with a value $r \in \{0, 1\}$ and sends it back to the challenger. u_a wins the game if:

$$\begin{cases} r = 0 \text{ and } b_j \leq b_k \text{ OR} \\ r = 1 \text{ and } b_j > b_k \end{cases}$$

The adversary u_a looses the game otherwise.

The coordinate-linkability advantage $Adv_a^{c-LNK}(A)$ is the probabilistic advantage of the adversary in winning this game:

$$Adv_a^{c-LNK}(A) = Pr[r = 0 \wedge b_j \leq b_k] + Pr[r = 1 \wedge b_j > b_k] - 1/2 \quad (7)$$

We can now define the *user-privacy* of any PPFRVP algorithm A as follows:

Definition 2: An execution of the PPFRVP algorithm A is user-private *if the identifiability advantage* $Adv_a^{IDT}(A)$*, the distance-linkability advantage* $Adv_a^{d-LNK}(A)$ *and the coordinate-linkability advantage* $Adv_a^{c-LNK}(A)$ *of each participating user* $u_i, i \in \{1, \ldots, N\}$ *are negligibly small.*

According to Definition 2, an execution of the PPFRVP algorithm is user-private if and only if any user u_a does not gain any (actually, negligible) additional knowledge about the preferred rendez-vous locations L_j of any other user u_j, $u_j \neq u_a$, except the final fair rendez-vous location L_{fair}.

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.

Definition 3: An execution of the PPFRVP algorithm A is server-private *if the identifiability advantage* $Adv_{LDS}^{IDT}(A)$*, the distance-linkability advantage* Adv_{LDS}^{d-LNK} *and the coordinate-linkability advantage* Adv_{LDS}^{c-LNK} *of an LDS are negligible.*

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. Thus, although it is critical to measure the privacy leakage of the PPFRVP algorithm in a single execution, it is also important to study the leakage that may occur over multiple *correlated* executions, which in turn depends on the intermediate and final output of the PPFRVP algorithm. We discuss the privacy leakage of the proposed algorithms over multiple executions in Section VI-D.

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. Privacy and Complexity Analysis

We first analyze the privacy of the proposed PPFRVP protocol (Fig. 4) with respect to the adversary model outlined in Section II-A.

A. Privacy Analysis Under Passive Adversary Model

Under the assumption of a passive adversary (both, LDS and participating users), we have the following result:

Proposition 1: The proposed PPFRVP protocols are correct and they guarantee identifiability- and coordinate-linkability privacy. However, they do not guarantee distance-linkability privacy.

Proof: Correctness: Given the encrypted set of user-preferred locations $f(L_1), \ldots, f(L_N)$, the proposed PPFRVP algorithms first compute the pairwise distance d_{ij} between each pair of users i and j, $\forall i, j \in \{1, \ldots, N\}$. One can easily verify that the ElGamal-Paillier-based distance computation algorithm computes:

$$\begin{aligned} Pai(d_{ij}^2) &= Pai(x_i^2) \cdot Pai(-2x_i x_j) \cdot Pai(y_j^2) \cdot Pai(y_i^2) \\ &\quad \cdot Pai(-2y_i y_j) \cdot Pai(y_j^2) \\ &= Pai(x_i^2 - 2x_i x_j + x_j^2 + y_i^2 - 2y_i y_j + y_j^2) \end{aligned}$$

The same result is achieved by the BGN-based distance algorithm.

After the pairwise distance computations, the PPFRVP algorithm computes the masking of these pairwise distances by scaling and shifting operations. The scaling operation is achieved by exponentiating the encrypted element to the power of r_i, where $r_i \in \mathbb{Z}_w^*$ is a random integer and r_i^{-1} is its multiplicative inverse. The shifting operation is done by multiplying the encrypted element with the encryption (using the public key of the users) of another random integer s_i privately chosen by the LDS. These two algebraic operations mask the values d_{ij}^2 (within the encrypted elements), such that the true d_{ij}^2 are hidden from the users. Nevertheless, thanks to the homomorphic properties of the encryption schemes, the LDS is still able to remove the masking (after the users have identified the maximum value) and correctly re-mask all maxima, such that each user is able to correctly find the minimum of all maxima.

In the end, each user is able to determine L_{fair}, where $fair = \text{argmin}_i \max_j d_{ij}^2$ from the outputs of the PPFRVP algorithm, and therefore the PPFRVP algorithms are correct.

1) User Identifiability Advantage: Hereafter we provide sketches of the proofs of user-privacy, after a private execution of the PPFRVP algorithm A. A sketch is usually given to intuitively show how the formal proof can be constructed with the argument presented in the sketch. In particular, the following sketches are exhaustive, i.e., they cover all possible cases, and they are used to show whether the different advantages are non-negligible and thus whether a PPFRVP algorithm A is execution privacy-preserving.

In the identifiability advantage, there are only two possible outcomes of the PPFRVP algorithm, depending on users' preferred locations L_i: The first case is when $L_{fair} = L_a$, i.e., when the fair rendez-vous location is the one proposed by the adversary; the second case is when $L_{fair} \neq L_a$, i.e., when the fair location is different from the one proposed by the adversary. Hereafter we split the sketch of our proof according to these two (and only possible) cases, and show that the advantage of the adversary is negligible in both these cases:

1) $\mathbf{L_{fair} = L_a}$: In this case, the adversary does not learn any additional information that was not already known to him before the execution of the protocol, except the order among the maximum distances between the users and the corresponding indices. Moreover, we consider here the non-trivial case where the challenger chooses a value $k \neq a$, otherwise the correct answer to the challenge is trivial. It should be noted that the challenger cannot select the trivial case with a probability greater than 1/N (during the challenge step or step 3). In this non-trivial case, the adversary cannot guess the value $k \neq a$ with a higher certainty than he would by a random guess because only the LDS knows the secret scaling and shifting values used for the masking operation. In fact, the order among the masked distances does not reveal any additional information about the actual locations, as there could be infinitely many locations at the same masked distance. Thus, the advantage of the adversary in this case is negligible.
2) $\mathbf{L_{fair} \neq L_a}$: In this case, the adversary learns, after the execution of the protocol, another preferred location $L_{fair} \neq L_a$ different from his own, in addition to the order among the maximum distances for all users. The adversary is able to compute the distance $d_{a,fair}$ between his preferred location and L_{fair}. However, thanks to the masking operation on the distances and to the independence among the users' preferred locations, the adversary has no additional knowledge to link $d_{a,fair}$ to any other masked d_i^{MAX} he knows. For instance, it is impossible for him to even compare $d_{a,fair}$ to any of the d_i^{MAX} as only the LDS knows the secret scaling and shifting values used for the masking operation. Hence, even with the additional knowledge of the $d_{a,fair}$ and L_{fair}, the adversary cannot guess the value of k with a

probability higher than a random guess. Thus, the advantage of the adversary is negligible in this case as well.

Considering the previous arguments, we have the following:

$$\begin{aligned} Adv_a^{IDT}(A) &= Pr(k' = k | L_{fair} = L_a) Pr(L_{fair} = L_a) \\ &\quad + Pr(k' = k | L_{fair} \neq L_a) Pr(L_{fair} \neq L_a) \\ &\quad - 1/N \\ &= 1/N \cdot 1/(N+1) + 1/N \cdot N/(N+1) - 1/N \\ &= 1/N - 1/N = 0 \end{aligned}$$

thanks to the independence of k' conditioned on the outcome L_{fair}. Thus, the identifiability-advantage is negligible.

2) User Coordinate-Linkability Advantage: Similarly to the identifiability advantage, there could only be two possible outcomes of any PPFRVP algorithm A, represented by the two cases $L_{fair} \neq L_a$ and $L_{fair} = L_a$. Hereafter we show that the advantage of the adversary is negligible in both cases.

1) $L_{fair} = L_a$: In this case, the adversary does not learn any additional information about the coordinates of any two users j, k. As the masked and ordered distances cannot be linked to a specific coordinate with a success probability higher than 1/3, the adversary cannot guess whether the coordinate value b_j is larger or smaller than b_k with a probability higher than a random guess (1/2). In fact, as the order among the masked distances is a relative measure between locations that is position-independent, it does not provide any additional information about the values of the coordinates of L_j, L_k. Thus, the advantage of the adversary is negligible.
2) $L_{fair} \neq L_a$: In this case, the adversary can once again compute the distance $d_{a,fair}$ between L_{fair} and L_a. As the distance by itself conveys no information about the orientation or relative position between L_j and L_k, $\forall j, k \in \{1, \ldots, N\}$ and $j \neq k$, the adversary cannot guess whether the coordinate b, randomly chosen by the challenger, is larger or smaller for L_j with respect to L_k with a higher certainty than a random guess. Thus, his advantage is negligible.

Similarly to the identifiability advantage, we obtain:

$$\begin{aligned} &dv_a^{c-LNK}(A) \\ &\quad = Pr(r = 0 \wedge b_j \leq b_k | L_{fair} = L_a) Pr(L_{fair} = L_a) \\ &\qquad + Pr(r = 0 \wedge b_j \leq b_k | L_{fair} \neq L_a) Pr(L_{fair} \neq L_a) \\ &\qquad + Pr(r = 1 \wedge b_j > b_k | L_{fair} = L_a) Pr(L_{fair} = L_a) \\ &\qquad + Pr(r = 1 \wedge b_j > b_k | L_{fair} \neq L_a) Pr(L_{fair} \neq L_a) \\ &\qquad - 1/2 \\ &\quad = Pr(r = 0) \cdot Pr(b_j \leq b_k) + Pr(r = 1) \\ &\qquad \cdot Pr(b_j > b_k) - 1/2 = 1/4 + 1/4 - 1/2 = 0 \end{aligned}$$

Thanks to the independence of the coordinate b from the outcome L_{fair}. Thus, the coordinate-linkability is negligible.

3) User Distance-Linkability Advantage: The PPFRVP algorithm defined in this manuscript takes as inputs the preferred rendez-vous locations L_i of each user $u_i \in \mathbb{U}$ and outputs both $f(L_{fair})$ and the set of randomized (but order-preserving) maximum distances $d_i^{max}, \forall u_i \in \mathbb{U}$. By means of an example, we show that there is at least one case in which our PPFRVP algorithm does not satisfy distance-linkability.

Suppose that, at Step 3 of the distance-linkability game, $\mathbb{C}$ chooses a value $s > \max_{u_i \in \mathbb{U}} d_i^{max}$. At Step 4, u_a obtains (s, j, k) and it knows that s is larger than any of the maximum randomized distances that it already possesses. Moreover, u_a also knows that the order-preserving randomization procedure $Rand(.)$ is such that $d_i^{max} = Rand(c_i^{max}) > c_i^{max}$, i.e., the randomization strictly increases the output compared to the input because the two randomizing factors r_i, s_i are positive. Hence, if $s > \max_{u_i \ \mathbb{U}} d_i^{max}$, u_a knows that for sure $s > d_{j,k}, \forall j \neq k$. Thus, u_a can win the game with non-negligible probability by choosing $s^* = 0$, proving that in this case our PPFRVP algorithm A does not satisfy user distance-linkability.

4) Third-Party Advantages: All elements that are received and processed by the LDS have previously been encrypted by the users with their common public key. In order to efficiently decrypt such elements, the LDS would need to have access to the private key that has been generated with the public key used for the encryption. As explained in Section II, in most practical settings, where service providers have a commercial interest in providing a faithful service to their customers, the LDS would not try to maliciously obtain the secret key. Therefore, all the LDS does in the PPFRVP algorithm is to obliviously execute algebraic operation on encrypted elements, without knowing the values within the encrypted elements. Hence, the PPFRVP algorithms do not disclose any information to third-parties, such as the LDS, during or after its execution. □

B. Privacy Analysis Under Active Adversary Model

We consider three main types of active attacks, namely (i) Collusion among users and/or LDS, (ii) Fake user generation and/or replay attacks and (iii) Unfair rendez-vous location.

1) Collusion: In the case of collusion among users, the published fair result can be used to construct exclusion zones. An exclusion zone is a region that does not contain any location preferences, and the number of such exclusion zones increases with the number of colluders. A set of colluding users could also select preferences which are close to each other, thus increasing the probability that the selected L_{fair} is one among these preferences. Similarly, the colluding users could select preferences far away from each other, so that L_{fair} is always selected from among the preferences of non-colluding users, thus revealing them. A much more serious case is the collusion between the LDS and a participant; the LDS could obtain the secret key shared by the participants, and thus learn the preferences of all the participants. These participants' preferences could be then shared by the LDS with the colluding user. The proposed PPFRVP protocols do not protect against such strong collusion attacks.

2) Fake Users: In case the LDS generates fake users, it would not be able to obtain the secret that is shared among the honest users and which is used to derive the secret key $K_s^{M_v}$ for each session v. This attack is more dangerous if a legitimate participant creates a fake, because the legitimate participant knows the shared secret. In this scenario, however, the LDS knows the list of meeting participants (as it computes the fair rendez-vous location) and therefore it would accept

TABLE I
ASYMPTOTIC COMPLEXITY ANALYSIS

CLIENT	PROTOCOL	BGN	ELGAMAL-PAILLIER
Multiplication	Distance	*O(1)*	*O(N)*
	PPFRVP		
Exponentiation	Distance	*O*(1)	*O(N)*
	PPFRVP	$O(N\sqrt{T})$	
Memory	Distance	*O(1)*	*O(N)*
	PPFRVP	*O(N)*	
Communication	Distance	*O(1)*	*O(N)*
	PPFRVP	*O(N)*	
LDS			
Multiplication, exponentiation	Distance	$O(N^2)$	$O(N^2)$
	PPFRVP		
Bilinear mapping	Distance	$O(N^2)$	-------
	PPFRVP		
Memory	Distance	$O(N^2)$	$O(N^2)$
	PPFRVP		
Communication	Distance	*O(N)*	$O(N^2)$
	PPFRVP	$O(N^2)$	

only messages digitally signed by each one of them. Here we rely on the fact that fake users will not be able to get their public keys signed by a CA. Replay attacks could be thwarted by verifying an individually signed *nonce*, derived using the shared secret, in each user's message.

3) Unfair RV: The last type of active attack could result in the computation of an unfair rendez-vous location. Malicious modification or untruthful reporting of the maximum masked values (Step B.3 of Fig. 4) could deceive the LDS in accepting a false received index as the maximum value, and therefore lead to the computation of a non-fair rendez-vous location. However, this is unlikely to happen in practice. For instance, even if in Step B.3 a user falsely reports one of his values to be the maximum, this would cause the algorithm to select a non-fair rendez-vous location if and only if no other user selected a smaller value as the maximum distance.

C. Complexity Analysis

Table I summarizes the complexity results for the proposed protocols. In this table, the *Distance* protocol is the one used in the module A of Fig. 4, whereas PPFRVP includes modules A,B and C. We can see that the client complexity is $O(N)$, where N is the number of participants. However, there is a notable exception for the BGN-based scheme; the number of exponentiation required for a single decryption is $O(\sqrt{T})$ [22], where T is the order of the plaintext domain.

The LDS complexity for both protocols is $O(N^2)$, with the notable exception of BGN, where, in addition to multiplications and exponentiations, the schemes requires additional $O(N^2)$ bilinear mappings. These operations are required to support the multiplicative property of the BGN scheme.

D. Privacy Under Multiple Dependent Executions

As defined earlier, in a dependent execution of the PPFRVP protocol, all the involved parties possess information from the previous executions, in addition to the current input, output and intermediate data. It is clear that, due to the oblivious or blind nature of the computations, the privacy guarantees of the proposed PPFRVP protocols with respect to the LDS in dependent executions remains the same as that for independent executions. Furthermore, dependent executions in which the information across executions is completely uncorrelated (e.g., different set of users in each execution or different and unrelated preferences in each execution) reduce to independent execution. We analyze two different scenarios of dependent executions involving differential information.

First, we consider the case of dependent executions with different subsets of participants. We assume that, in each sequential execution, the set of users or participants is reduced by exactly one (the adversary participant remains until the end), and that the retained participants preferences remain the same as the previous execution(s). The following information is implicitly passed across executions in this scenario: (i) participant set, (ii) optimal fair location L_{fair}, (iii) permuted and randomly scaled pairwise distances from the participant to every other participant, and (iv) scaled (but order preserving) maximum distance from every participant to every other participant. One observation we can make is that, as participant preferences do not change, no new or additional information is available to the adversary from one execution to the next. In this scenario, one trivial attack on identifiability that the adversary can carry out is as follows. As the user corresponding to L_{fair} is known in each execution, the adversarial participant can repeatedly execute the PPFRVP protocol with a subset of users by excluding the user(s) corresponding to L_{fair} in the previous executions. This way, in each execution, the adversary learns the location preference (L_{fair}) of a different participant. An LDS can prevent such attacks by keeping some memory of past executions and not allowing future executions with small differences in the participant sets. A notion of (k, t)-differential privacy could be defined wherein the LDS will not execute the protocol on a new request if there was an execution recorded at most t time units earlier consisting of all the participants as present in the current request and at most k additional (and different) participants.

Second, we consider the case of dependent executions with always the same participants, where they can change their preferred locations at each execution. In the worst case scenario, the preferences of all participants (except the adversary) are fixed across executions. Due to the scaled (and permuted) pairwise distances, as well as the scaled maximum distances, the adversary cannot identify specific locations of users (except the user corresponding to L_{fair}), but he could use the scaled distance information in the previous rounds to manipulate his own location preference in the current execution to gain an unfair advantage. As the LDS is oblivious to participant inputs, and it cannot detect and prevent this type of data manipulation attack. This type of attack can be prevented by the honest participants (or users) by refusing to participate with same preferences in multiple executions involving the same set of participants.

VII. EXPERIMENTAL EVALUATION

In this section, we present an in-depth evaluation of the proposed PPFRVP protocols by outlining the results of

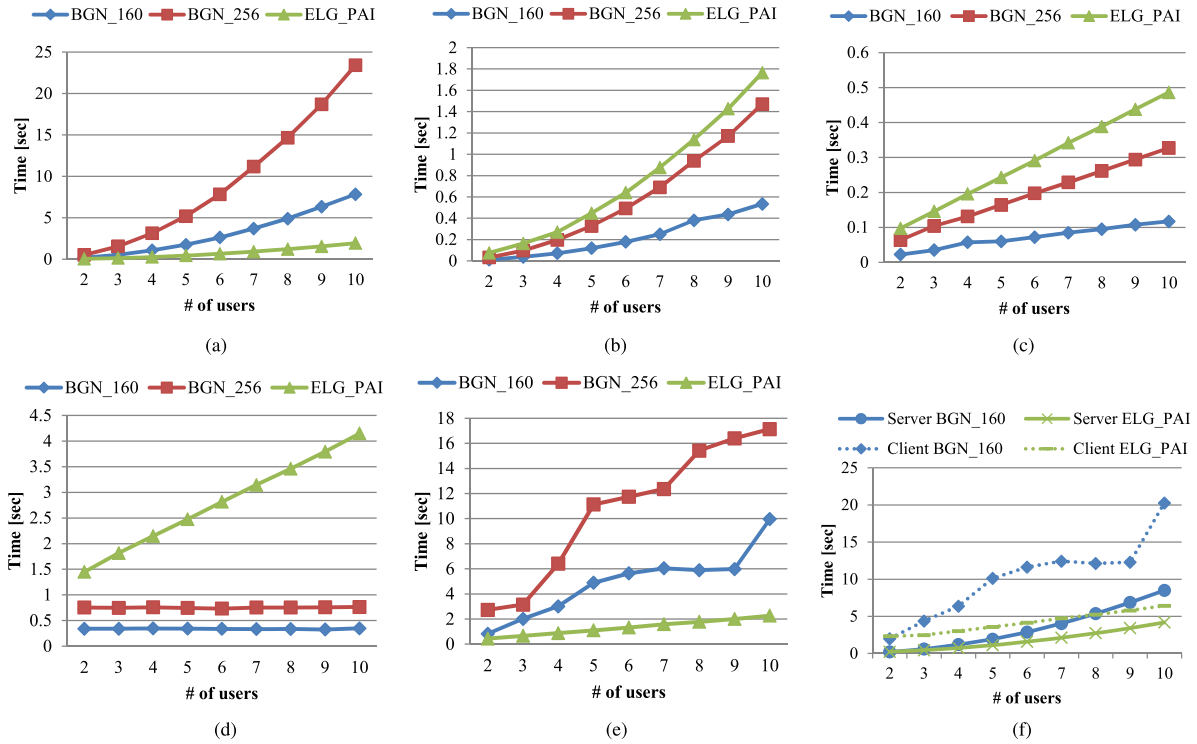


Fig. 5. Performance measurements. (a) LDS distance computations (module A). (b) LDS maximum computations (module B). (c) LDS minimum computations (module C). (d) Client distance computations (module A). (e) Client max/argmin computations (module B/C). (f) Total client and LDS run times (modules A+B+C).

controlled experiments and user studies conducted using prototype implementation of the protocols on modern mobile devices.

A. Implementation and Performance Measurements

The client application is implemented on Nokia N810 mobile devices (ARM 400 MHz CPU, 256 MB RAM, Linux Maemo OS) and the LDS implementation is running on a standard Linux PC (2 GHz CPU, 3 GB RAM, Ubuntu Linux). Our applications are implemented using the Qt programming framework.

For 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 [25] cryptosystems.

1) Computation Delay on the LDS: We can see Fig. 5(a), 5(b) and 5(c) that 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 to execute 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 (Fig. 5(b) and 5(c)). 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: Fig. 5(d) and 5(e) show 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, we can see that the ElGamal-Paillier based protocol has a better performance. Nevertheless, both schemes perform reasonably well on current generations of mobile devices. It is also important to observe that the results obtained in our experiments are based on our prototype implementation of the BGN scheme, which is not optimized for performance.

3) Power Consumption Analysis on the Nokia N810 Clients: In order to analytically evaluate the power consumption of the PPFRVP protocol computations, we utilize the power model proposed by Kaneda et al. [26]. 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. The general power model is outlined by the following polynomial expression (Eqn. 8), where the Nokia N810 specific coefficients c_i are determined by using a multiple regression technique.

$$P_{est} = c_0 + c_1 \cdot P_{cpu}^{c_2} + c_3 \cdot P_{recep}^{c_4} + c_5 \cdot P_{trans}^{c_6} + c_7 \cdot P_{wact}^{c_8} + c_9 \cdot P_{read}^{c_{10}} + c_{11} \cdot P_{write}^{c_{12}} \quad (8)$$

In Eqn. 8, P_{est} is the estimated per second power consumption, and P_{cpu}, P_{recep}, P_{trans}, P_{wact}, P_{read} and P_{write} represent the per second power consumption due to CPU utilization, wireless data reception throughput, wireless data transmission throughput, wireless LAN activity period, and throughput of data read and write from a flash disk, respectively. In the above power model, Kaneda et al. [26] showed that it is difficult to derive the coefficients c_3, c_4, c_5, c_6, c_7 and c_8 accurately because of the strong correlation between P_{trans} and P_{wact}, and between P_{recep} and P_{wact}. Moreover, they also showed that the power consumption of tasks that mostly utilize the CPU (without utilizing WLAN) can be more accurately determined using the above model. As currently we are interested in only measuring power consumption due to protocol computations, we can set P_{recep}, P_{trans} and P_{wact} to zero. Also, as we never store or read data from the N810 flash disk in our experiments, we can set P_{read} and P_{write} to zero. Thus, Kaneda et al.'s power model [26] for a Nokia N810 by considering only CPU utilization can be outlined as:

$$P_{est} = 0.4650 + 0.5910 \cdot (P_{cpu})^{1.0472} \quad (9)$$

Now, given the above model, we determine the power consumed by the computations of our proposed PPFRVP protocols as follows. From Fig. 5(f), we can see that the total computation time of the BGN-based protocol (with a 160-bit key) for 5 participants on a N810 device is 10 seconds, whereas, for the ElGamal-Paillier-based protocol it is approximately 4 seconds. Assuming 100% CPU utilization (i.e., $P_{cpu} = 1$), the computation of the BGN-based protocol with 5 participants is expected to consume approximately $10.56W$ on each client, whereas, the ElGamal-Paillier-based protocol would consume roughly $4.056W$. Readers should note that this power consumption only includes the CPU utilization and does not consider the data transfer operations (and the related CPU computations) of the protocols. Nokia N810s come equipped with a 1500 mAh BP-4L battery operating at $3.7V$, which means it can drive the a system consuming $5.5W$ for roughly an hour, or equivalently a system consuming $1.1W$ for 5 hours (18000 seconds). This means that a fully charged Nokia N810 could roughly support 1800 such BGN-based or 4500 such ElGamal-Paillier-based PPFRVP protocol executions, not considering the data transfer and WLAN operations, as well as other applications running on the system. In practice, we expect the actual number of protocol executions supported with a single charge to be significantly lower, considering the aggregated consumption of other system components.

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. The questionnaires are based on the privacy and usability guidelines from [27] and [28].

1) User-Study: The user study consists of three phases: 1) 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.

Phase 2: In this phase, participants used our PPFRVP prototype application multiple times in seperate groups. The participants were seated individually and they could not speak to other participants or see their device screens.

Phase 3: Fig. 6 summarizes the main findings after the users interacted with our prototype. All participants tend to agree (34%) or agree (66%) that it was easy to use, and that they could quickly compute the task (97%). Around 80% feel that it was easy to learn to use such application, and 91% tend to agree or agree that the GUI was clearly organized. More than 68% agree that the application was interesting to use, and if we include those who *tend to agree* as well, all but one participant found it interesting. With respect to privacy in such applications, 66% agree that it is important to reveal only the necessary information to the system. More than 71% appreciated that their preferred rendez-vous point was not revealed to other participants, and only 8% did not care about

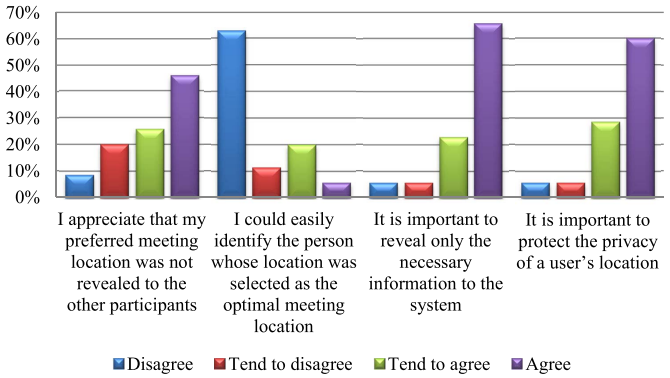


Fig. 6. Summary of the user-study results for Phase 3.

the privacy of their rendez-vous location preference. Only 26% of the participants were able to identify to whom the FRVP location belonged to, which was to be expected. The users ran our application in groups of 5 during the experimentation, and therefore there was always one person out of five that knew that the FRVP location was his preferred location. From an application standpoint, these results mean that both ease-of-use and privacy are important in LSBS. In particular, the privacy mechanisms should be implemented in a way that does not significantly affect the usability or performance. Moreover, the acceptance of LSBS applications is highly influenced by the availability of effective and intuitive privacy features.

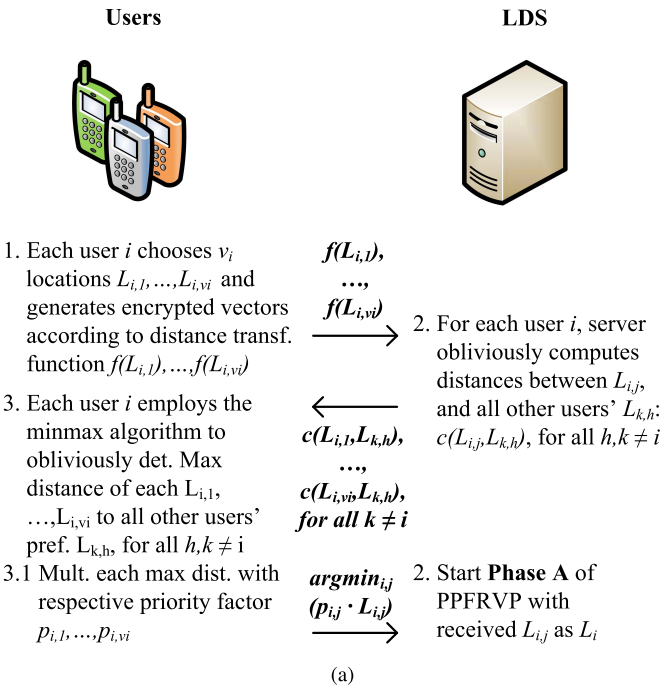
VIII. Extension to Multiple Preferences

The PPFRVP protocol, as defined in Section III, 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 $L_{i,1}, \ldots, L_{i,vi}$. 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. Fig. 7(a) outlines the PPFRVP protocol with multiple location preferences.

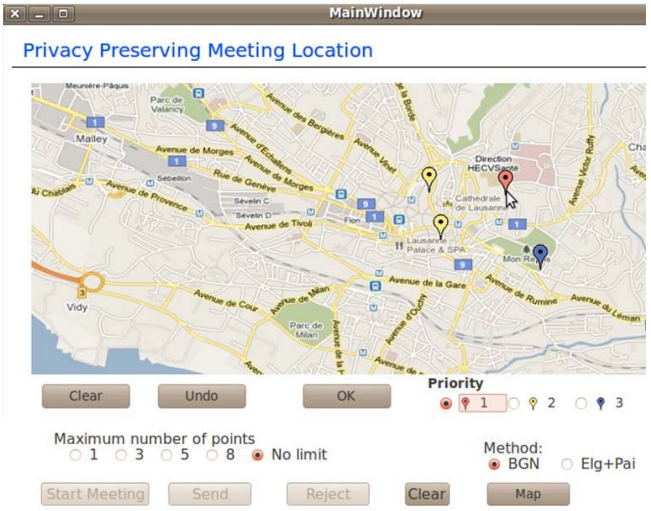
Multiple preferences are included in the PPFRVP protocol by assigning a priority to each preferred location $L_{i,j}$ for all users i and preferences j. One way to include them in the distance computations is to assign weighting coefficients $p_{i,j}$ for the maximum distances $c(L_{i,j}, L_{k,h})$ computed at the end of Step 3; this way, the highest priority could be defined by using the lowest value of $p_{i,1} = 1$, whereas the lower priorities could be assigned higher values of $p_{i,2} = p_{i,3} = 2$ (as in Fig. 7(b)). As a result, the minimum of these maximum meta-distance 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.

A. Privacy Discussion

The proposed extension to the PPFRVP protocol relies on the same cryptographic primitives as the original PPFRVP protocol. As the operations performed by the LDS are essentially the same, the extended PPFRVP protocol also inherits the privacy guarantees possessed by the single-location PPFRVP one. Therefore, Proposition 1 holds for the PPFRVP protocols (BGN- and ElGamal-Paillier-based) with



(a)



(b)

Fig. 7. Extension of PPFRVP to multiple user-preferred locations. (a) Extension of the existing PPFRVP protocol. (b) Prototype application.

the extension to multiple user-preferred locations in the passive adversary scenario. However, it retains the same vulnerabilities in the active adversary scenario.

B. Performance Discussion

As compared to the ElGamal-Paillier-based protocol, the BGN-based distance computations of the extension clearly reduce the number of message exchanges between each client and the LDS. However, as there is a decryption operation performed by the clients in Step 3, the distance computation for the extended protocol with the BGN scheme increases the overall complexity of the protocol to $O(N\sqrt{T}\max_i vi)$, where vi is the number of location preferences of user i. Compared to this, the overall complexity of the ElGamal-Paillier-based scheme is $O(N \max_i vi)$.

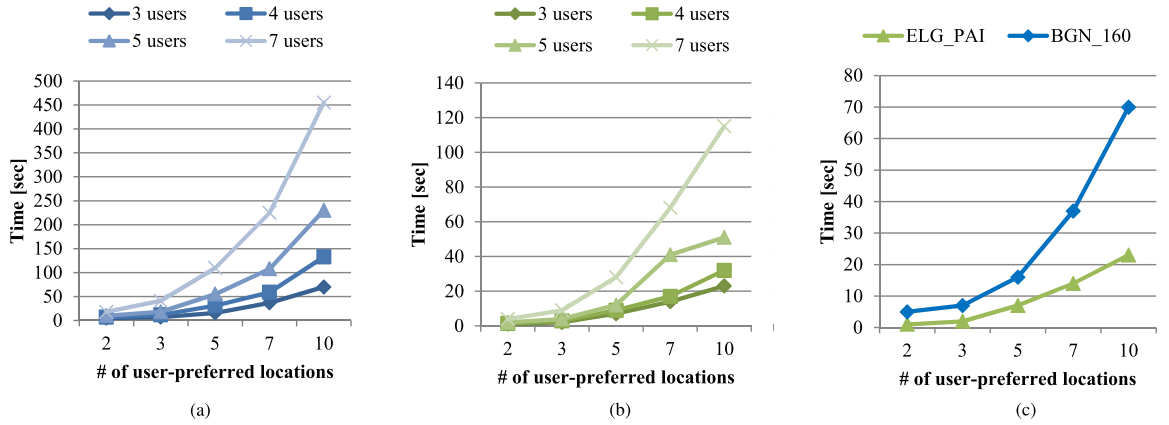


Fig. 8. LDS processing times for the PPFRVP multi-preference extension. (a) BGN-based protocol using a 160-bit key. (b) ElGamal-Paillier-based protocol. (b) Comparison.

The communication complexity would however remain the same for both protocols, which is $O(N \max_i vi)$. Hence, the ElGamal-Paillier-based, extended PPFRVP protocol would be more preferable from a performance standpoint.

C. Implementation and Evaluation

We have implemented the proposed PPFRVP algorithm for handling multiple location preferences by extending our earlier application that takes a single preference per user. A screen-shot of the application front-end is shown in Fig. 7(b).

We also perform a series of controlled experiments in order to evaluate the performance of our PPFRVP multi-preference extension. In the first set of experiments, we measure the performance of our BGN-based PPFRVP multi-preference extension algorithm on the Nokia N810 mobile devices. For this set of experiments, we use the BGN cryptosystem with a 160-bit key. For three N810 clients and four location preferences per client, we observe an average execution time of around 73 seconds on the client. This delay is almost 15 times larger than the corresponding single preference case. The results for the BGN-based protocol with a 256-bit key and the ElGamal-Paillier-based multi-preference extension shows a similar trend, with occasional client-side application failures. We attribute these large client-side execution delays and failures to the hardware limitations of the Nokia N810 in performing extensive cryptographic operations, including exponentiation operations of the Paillier cryptosystem and the bilinear mapping operations involved in the BGN scheme.

Due to the hardware limitations of Nokia N810, we evaluate the LDS execution times for our PPFRVP multi-preference extension by executing the clients on Nokia Maemo emulators [29]. Each client application is executing on a separate emulator instance, but all emulator instances are running on the same workstation (a standard workstation with 2 GHz CPU, 3 GB RAM and running Ubuntu Linux). This workstation (running client emulators) makes a wireless connection to the LDS workstation using standard WiFi in the ad hoc mode. We evaluate the overall processing times (excluding communication time) of our BGN-based protocol using a 160-bit key and the ElGamal-Paillier-based protocol. For both these evaluations, we execute our protocol under different scenarios containing an increasing number (3, 4, 5, and 7) of users with an increasing (2, 3, 5, 7, and 10) number of location preferences per user. The results of these evaluations are outlined in Fig. 8.

We can see from Fig. 8(a) and 8(b) that the LDS processing time for both the BGN-based and the ElGamal-Paillier-based protocols increases as the number of users and the number of location preferences per user increases, which is very intuitive. One other observation we make is that the LDS processing time grows much faster as the number of location preferences increases. For example, for 7 users using the BGN-based protocol with 160-bit key, the LDS processing delay jumps from 5 seconds for a single preference to around 225 seconds for seven preferences per user. Similarly, the ElGamal-Paillier-based protocol sees a jump from a few seconds to almost 70 seconds. Fig. 8(c) shows a comparison between the LDS processing times of the ElGamal-Paillier-based protocol and the BGN-based protocol for three users or clients. We can see that, similar to the single preference case, the ElGamal-Paillier-based protocol generally outperforms the BGN-based protocol with respect to the LDS processing time. As the number of location preferences per user increases, the LDS processing time of BGN-based protocol grows much faster compared to the ElGamal-Paillier-based protocol. Thus, ElGamal-Paillier-based protocol is preferred over the BGN-based protocol for a large number of location preferences.

Based on the experimental results presented in this section, we can conclude that, although the multi-preference extensions of the PPFRVP algorithms are feasible (at least for lower number of users/preferences), the high delays, both at the clients and at the LDS, make them unattractive from the usability perspective. One of the main reason for this is the inability of existing portable mobile device hardware to efficiently carry out complex cryptographic operations. As our current evaluations are solely based on the performance of the proposed schemes on the Nokia N810 class of mobile devices, one of the main task that we want to accomplish in the future is to conduct a thorough evaluation of our schemes on a variety of latest mobile hardware and software platforms. In addition to this, one of our future goal is to improve existing mechanisms, and propose alternative ones if needed,

in order to make them more efficient in handling the multiple-preference scenarios.

IX. Related Work

The problem of privacy-preserving fair rendez-vous location has received little or no attention in the literature. Santos and Vaughn [30] present a 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, Berger et al. [31] propose 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 [32] or points [33], [34]. Frikken and Atallah [32] propose 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. Zhong et al. [35] 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 check-ins*. However, these research efforts attempt to solve issues that are orthogonal, and uniquely different, from the ones addressed in this paper. Jaiswal and Nandi [36] propose a privacy-preserving platform, called *Trust No One*, for privately locating nearby points-of-interest. Their architecture relies on three non-colluding parties, i.e., the mobile operator, the LBS provider, and the matching service, for decoupling user locations from user queries. The architecture proposed by Jaiswal and Nandi [36] addresses the problem of location-privacy preserving information retrieval, which is different from our focus.

In the direction of anonymous location sharing, Pidcock et al. [40] propose 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 [40], Guha et al. [41] propose a privacy-preserving system that allows users to set location-triggered alarms based on presence at specific locations, rather than sharing location coordinates. Carbunar et al. [43] 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.

X. 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] J. 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.

[9] J. Freudiger, M. Jadliwala, J.-P. Hubaux, V. Niemi, P. Ginzboorg, and I. Aad, "Privacy of community pseudonyms in wireless peer-to-peer networks," *Mobile Netw. Appl.*, vol. 18, no. 3, pp. 413–428, 2012.

[10] (2011, Nov.). *Please Rob Me* [Online]. Available: http://pleaserobme.com/

[11] J. Krumm, "A survey of computational location privacy," *Personal Ubiquitous Comput.*, vol. 13, no. 6, pp. 391–399, 2009.

[12] V. Vazirani, *Approximation Algorithms*. New York, NY, USA: Springer-Verlag, 2001.

[13] I. Bilogrevic, M. Jadliwala, K. Kalkan, J. Hubaux, and I. Aad, "Privacy in mobile computing for location-sharing-based services," in *Proc. 11th Int. Conf. PETS*, 2011, pp. 77–96.

[14] (2011, Nov.). *UTM Coordinate System* [Online]. Available: https://www.e-education.psu.edu/natureofgeoinfo/c2_p21.html

[15] G. Ghinita, P. Kalnis, A. Khoshgozaran, C. Shahabi, and K. Tan, "Private queries in location based services: Anonymizers are not necessary," in *Proc. ACM SIGMOD*, 2008, pp. 121–132.

[16] M. Jadliwala, S. Zhong, S. J. Upadhyaya, C. Qiao, and J.-P. Hubaux, "Secure distance-based localization in the presence of cheating beacon nodes," *IEEE Trans. Mobile Comput.*, vol. 9, no. 6, pp. 810–823, Jun. 2010.

[17] C.-H. O. Chen *et al.*, "GAnGS: Gather, authenticate 'n group securely," in *Proc. 14th ACM Int. Conf. Mobile Computing Networking*, 2008, pp. 92–103.

[18] Y.-H. Lin *et al.*, "SPATE: Small-group PKI-less authenticated trust establishment," in *Proc. 7th Int. Conf. MobiSys*, 2009, pp. 1–14.

[19] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Commun. ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[20] O. Goldreich, *Foundations of Cryptography: Basic Applications*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[21] A. Loukas, D. Damopoulos, S. A. Menesidou, M. E. Skarkala, G. Kambourakis, and S. Gritzalis, "MILC: A secure and privacy-preserving mobile instant locator with chatting," *Inf. Syst. Frontiers*, vol. 14, no. 3, pp. 481–497, 2012.
[22] D. Boneh, E.-J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proc. TCC*, 2005, pp. 325–341.
[23] T. ElGamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," *IEEE Trans. Inf. Theory*, vol. 31, no. 4, pp. 473–481, Jul. 1985.
[24] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. 17th Int. Conf. Theory Application Cryptographic Techniques*, 1999, pp. 223–238.
[25] M. Robshaw and Y. Yin, "Elliptic curve cryptosystems," RSA Lab., Bedford, MA, USA, Tech. Rep., 1997.
[26] Y. Kaneda, T. Okuhira, T. Ishihara, K. Hisazumi, T. Kamiyama, and M. Katagiri, "A run-time power analysis method using OS-observable parameters for mobile terminals," in *Proc. ICESIT*, 2010, pp. 1–6.
[27] M. Chignell, A. Quan-Haase, and J. Gwizdka, "The privacy attitudes questionnaire (PAQ): Initial development and validation," in *Proc. Human Factors and Ergonomics Society Annu. Meeting*, 2003.
[28] J. Lewis, "IBM computer usability satisfaction questionnaires: Psychometric evaluation and instructions for use," *Int. J. Human Comput. Interact.*, vol. 7, no. 1, pp. 57–78, 1995.
[29] (2013, Dec.). *Nokia N900 Maemo Emulator* [Online]. Available: http://doc.qt.digia.com/qtcreator-2.1/creator-maemo-emulator.html
[30] P. Santos and H. Vaughn, "Where shall we meet? Proposing optimal locations for meetings," in *Proc. MapISNet*, 2007.
[31] F. Berger, R. Klein, D. Nussbaum, J.-R. Sack, and J. Yi, "A meeting scheduling problem respecting time and space," *GeoInformatica*, vol. 13, no. 4, pp. 453–481, 2009.
[32] K. B. Frikken and M. J. Atallah, "Privacy preserving route planning," in *Proc. ACM WPES*, 2004, pp. 8–15.
[33] S.-D. Li and Y.-Q. Dai, "Secure two-party computational geometry," *J. Comput. Sci. Technol.*, vol. 20, no. 2, pp. 258–263, 2005.
[34] A. Solanas and A. Martínez-Ballesté, "Privacy protection in location-based services through a public-key privacy homomorphism," in *Proc. 4th European Conf. Public Key Infrastructure, Theory and Practice*, 2007, pp. 362–368.
[35] G. Zhong, I. Goldberg, and U. Hengartner, "Louis, Lester and Pierre: Three protocols for location privacy," in *Proc. 7th Int. Conf. Privacy Enhancing Technologies*, 2007, pp. 62–76.
[36] S. Jaiswal and A. Nandi, "Trust no one: A decentralized matching service for privacy in location based services," in *Proc. ACM MobiHeld*, 2010.
[37] B. Gedik and L. Liu, "Location privacy in mobile systems: A personalized anonymization model," in *Proc. 25th IEEE ICDCS*, Jun. 2005, pp. 620–629.
[38] C. Ardagna, M. Cremonini, E. Damiani, S. Vimercati, and P. Samarati, "Location privacy protection through obfuscation-based techniques," in *Proc. 21st IFIP WG 11.3 Working Conf. Data and Applications Security*, 2007.
[39] C. Zhang and Y. Huang, "Cloaking locations for anonymous location based services," *GeoInformatica*, vol. 13, no. 2, pp. 159–182, 2009.
[40] S. Pidcock and U. Hengartner, "Zerosquare: A privacy-friendly location hub for geosocial applications," in *Proc. 2nd ACM SIGCOMM Workshop Networking, Systems, and Applications Mobile Handhelds*, 2013.
[41] S. Guha, M. Jain, and V. Padmanabhan, "Koi: A location-privacy platform for smartphone apps," in *Proc. 9th USENIX Conf. NSDI*, 2012.
[42] M. Herrmann, A. Rial, C. Diaz, and B. Preneel, "Privacy-preserving location-sharing-based services," COSIC, Katholieke Univ. Leuven, Leuven, Belgium, Tech. Rep., 2013.
[43] B. Carbunar, R. Sion, R. Potharaju, and M. Ehsan, "The shy mayor: Private badges in geosocial networks," in *Proc. 10th Int. Conf. ACNS*, 2012, pp. 436–454.

Igor Bilogrevic is a Privacy Researcher with the Laboratory for Computer Communications and Applications, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, where he received the M.Sc. and Ph.D. degrees in communication systems in 2009 and 2014, respectively. His main areas of interest include privacy and security of context-aware mobile networks, applications of machine learning for privacy, contextual intelligence, data analytics, user behavior, and cellular networks, including femtocells.

Murtuza Jadliwala is currently an Assistant Professor with the Electrical Engineering and Computer Science Department, Wichita State University, Wichita, KS, USA, where he directs the Security, Privacy, Trust and Ethics in Computing Research Laboratory. Prior to that, he was a Senior Researcher with the Laboratory for Computer Communications and Applications, Swiss Federal Institute of Technology, Lausanne, Switzerland. He received the M.S. and Ph.D. degrees in computer science from the University at Buffalo, State University of New York, Buffalo, NY, USA, in 2004 and 2008, respectively.

Vishal Joneja received the master's degree in computer science from Ecole Polytechnique Fdrale de Lausanne, Lausanne, Switzerland, in 2011. During his master's project, he developed several applications and tools, such as a social and collaboration tool for business data analytics applications. The latter project was carried out was at Siemens Corporate Research, Inc., Princeton, NJ, USA. Since 2011, he has been a Consultant with ELCA Informatique SA, Lausanne, a Swiss IT-Solution Company. His professional interests include providing efficient IT solutions on demand from various companies in Europe and Switzerland, in particular.

Kübra Kalkan received the B.Sc. and M.Sc. degrees from the Department of Computer Science and Engineering, Sabanc University, Istanbul, Turkey, in 2009 and 2011, respectively. She visited Ecole Polytechnique Fdrale de Lausanne, Lausanne, Switzerland, as a Research Intern, which was supervised by Prof. Jean-Pierre Hubaux for four months. She started working toward the Ph.D. degree in 2011 at the Department of Computer Engineering, Bogazii University. During the summer of 2012, she went to Microsoft, Redmond, WA, USA, for an internship project. Now she is a research assistant with Bogazii University. Her current research interests are computer and network security, wireless networks, and cryptography.

Jean-Pierre Hubaux (F'08) is a full professor at EPFL. His current research activity is focused on privacy protection mechanisms, notably in pervasive communication systems and online social networks. He has recently started research activity in genomic privacy, in close collaboration with geneticists. He contributed some of the main ideas of the National Competence Center in Research named "Mobile Information and Communication Systems" (NCCR/MICS). In 2003, he identified the security of vehicular networks as one of the main research challenges for real-world mobile ad hoc networks. In 2008, he completed a graduate textbook titled Security and Cooperation in Wireless Networks, with L. Buttyan. He held visiting positions at IBM T.J. Watson Research Center and at UC Berkeley. Since 2007, he has been one of the seven commissioners of the Swiss FCC. He is a Fellow of ACM (2010).

Imad Aad is a Research Scientist with the University of Bern, Bern, Switzerland, where he is involved in mobile cloud networking and security in information centric networks. He received the M.Sc. and Ph.D. degrees from the University of Nice-Sophia Antipolis, Nice, France, and Institut National de Recherche en Informatique et en Automatique, France, in 1999 and 2002, respectively. He was a Post-Doctoral Researcher with Ecole Polytechnique Fdrale de Lausanne, Lausanne, Switzerland, from 2003 to 2005. He was a Senior Researcher with DOCOMO Euro-Labs, Munich, Germany, from 2005 to 2009, after which, he was with the Nokia Research Center, Zurich, Switzerland, until 2012. His research interests include privacy in contextual services, security, and multiple access control.