ISSN:0975-9646

A Secured Location Estimation Technique in Mobile Devices

T.Prabhakara Rao[#1], Kanugula Lalitha Kumari[*2]

[#]*Assistant.Professor,AITAM,Tekkali, India*
[*]*M.Tech, AITAM,Tekkali, India*

***Abstract*— 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.**

***Keywords*— Privacy, Mobile, Smartphones**

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 references, 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.

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-basedqueries* and *location sharing* or anonymous *location checkins*. However, these research efforts attempt to solve issues that are orthogonal, and uniquely different, from the ones addressed in this paper. 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.

SYSTEM ARCHITECTURE

We consider a system composed of two main entities: (i) a set of users (or mobile devices) $U = \{u_1, \ldots, u_N\}$ 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 $Li = (xi , yi) \in N2$ 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.

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

The mobile devices are able to perform public-key cryptographic operations. We assume that each of the N users has his own public/private key pair $(K^{ui}_P , K^{ui}s)$, 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^{Mv}_P , K^{Mv}_s) in an online fashion for each meeting setup instance v. The private key $K^{Mv}s$ generated in this way is known only to all meeting participants, whereas the public key K^{Mv}_P 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^{LDS}_P be the public key, certified by a trusted CA, and K^{LD0}_S the corresponding private key of the LDS. K^{LDS}_P 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^{LDS}_s . 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 $K_p{}^{LD}$, 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.

PPFRVP PROBLEM FORMULATION

In this work, we consider the problem of finding a rendezvous point among a set of user-proposed locations, such that (i) the rendez-vous point is *fair* (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-PreservingFair Rendez-Vous Point (PPFRVP)* algorithm.

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.

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

1) Distance Computation: The distance computation module uses either the BGN-distance or the Paillier-ElGamaldistance protocols. We note that modules (B) and (C) use the same encryption scheme as the one used in module (A). In other words, *E(.)* 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 $c^{tot}{}_{i\,j}$ (the encrypted square distance between L_i and L_j) for all j , thus, obtaining $d{*}i\,j$. 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 $d{*}i\,j$, 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 index $\sigma maxi$ of the maximum value to the LDS. 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: 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. Each user decrypts the received masked (randomly scaled and shifted) maximum values, and determines the minimum among all maxima. Each user knows which identifier corresponds to himself, and the user whose preferred location has the minimum distance sends to all other users the fair rendezvous location in an anonymous way. After the last step, each user receives the final fair rendez-vous location, but no other information regarding non-fair locations or distances is leaked.

PRIVACY AND COMPLEXITY ANALYSIS

We first analyze the privacy of the proposed PPFRVP protocol 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 (L1), . . . , f (LN), the proposed PPFRVP algorithms first compute the pairwise distance di j between each pair of users i and j , ∀i, j ∈ {1, . . . , 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 $r_i \in Z^*_w$ is a random integer and $r^{-1}{}_i$ 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 $d2_{i\,j}$ (within the encrypted elements), such that the true $d2_{i\,j}$ 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 $L_{f\,air}$, where $fair = \text{argmin}_i \max_j d2_{i\,j}$ 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 Li : The first case is when L f air = La, i.e., when the fair rendez-vous location is the one proposed by the adversary; the second case is when L f air _= La, 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) **Lfair = La**: 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.

Lfair ≠ La: 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_{MAXi} he knows. For instance, it is impossible for him to even compare $d_{a,\,fair}$ to any of the d_{MAXi} 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 $L_{f\ air}$. 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 $L\ fair \neq La$ and $L\ fair = La$. Hereafter we show that the advantage of the adversary is negligible in both cases.

L f air = La: 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 bk 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 $L\ j$, Lk. Thus, the advantage of the adversary is negligible.

Lfair ≠ La: In this case, the adversary can once again compute the distance *da, f air* between *L f air* and *La*. As the distance by itself conveys no information about the orientation or relative position between $L\ j$ and Lk , $\forall\ j,\ k \in \{1, \ldots, N\}$ 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 Lk with a higher certainty than a random guess. Thus, his advantage is negligible.

Similarly to the identifiability advantage, we obtain:

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

Thanks to the independence of the coordinate b from the outcome $L_{f\ air}$. 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 $u_i \in \mathsf{U}$ and outputs both $f\ (L_{f\ air})$ and the set of randomized (but orderpreserving) maximum distances d_{maxi} , $\forall u_i \in \mathsf{U}$. By means of an example, we show that there is at least one case in which our PPFRVP algorithm does not satisfy distance-linkability.

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.

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 Mvs* for each session *v*. This attack is more dangerous if alegitimate 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 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.

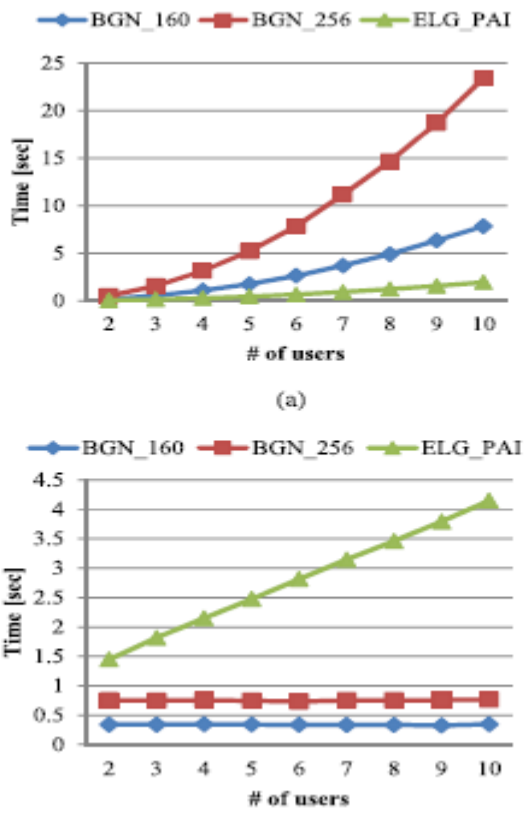

FIGURE 1: LDS DISTANCE COMPUTATIONS AND CLIENT DISTANCE COMPUTATIONS.

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. 1) 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.

EXPERIMENTAL EVALUATION

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

Devices 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.

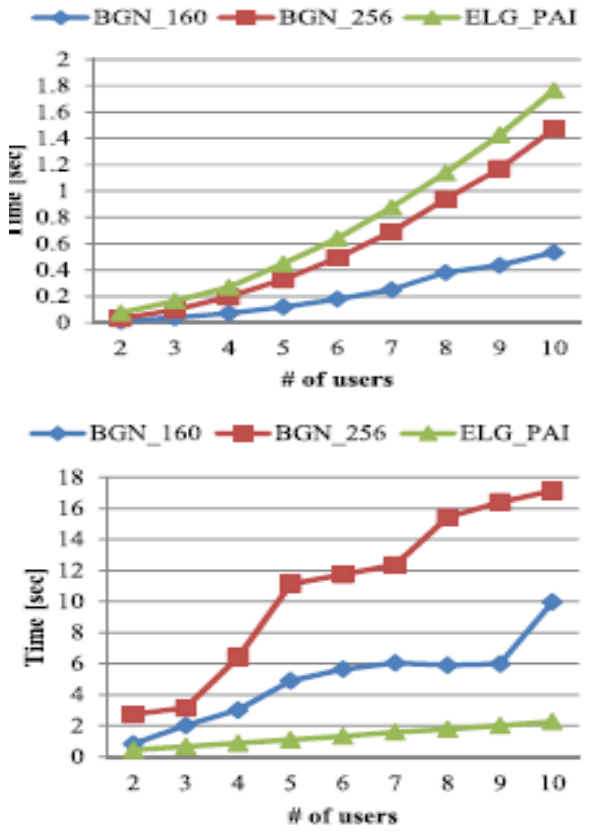

Figure 2: LDS maximum computations and Client max/argmin computations

1) Computation Delay on the LDS: We can see in the Figure 1 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.

2) Computation Delay on the Nokia N810 Clients: Figure 2 shows 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.

CONCLUSIONS

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.

ACKNOWLEDGMENT

The heading of the Acknowledgment section and the References section must not be numbered. Causal Productions wishes to acknowledge Michael Shell and other contributors for developing and maintaining the IEEE LaTeX style files which have been used in the preparation of this template. To see the list of contributors, please refer to the top of file IEEETran.cls in the IEEE LaTeX distribution.

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 privacypreserving 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 locationbased 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.