Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 00 (2014) 000–000

Procedia
Computer Science

www.elsevier.com/locate/procedia

The 11th International Conference on Mobile Systems and Pervasive Computing
(MobiSPC-2014)

Nearby Friend Discovery with Geo-Indistinguishability to Stalkers

Changsha Ma and Chang Wen Chen*

Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY, 14260, USA

Abstract

Nearby friend discovery is a popular location based service (LBS), which allows you to discover nearby like-minded people, and make friends with them. The main privacy threat of this service is that the disclosure of locations leaves opportunities for stalkers. Although location privacy preservation in LBS has recently received much attention, few works have been done on privacy-aware nearby friend discovery, where people want to discover nearby friends without exposing their private locations to arbitrary strangers. Unlike most of other LBS services, nearby friend discovery needs to consider the privacy of both of the two communicating entities, i.e., the user who is searching for nearby people, and the user who is being discovered by others. This unique property makes it a great challenge to protect users' location privacy while providing satisfactory service quality. This paper presents the first research addressing this issue by combining the location approximation technique and the homomorphic cryptography. We show that the proposed scheme provides formal privacy guarantees for the LBS users, and still achieves satisfactory quality of the LBS.

Selection and peer-review under responsibility of Elhadi M. Shakshuki.

Keywords: LBS; nearby friend discovery; privacy; geo-indistinguishability; homomorphic cryptography

1. Introduction

The prevalence of mobile devices equipped with GPS has enabled the geo-social networks, which support varied Location Based Services (LBS), to become increasingly popular. With the additional location information collected by the LBS servers, geo-social networks could provide the LBS users with unique services that are absent in current social networks. Nearby friend discovery is a popular LBS that is provided in many geo-social networks, such as Loopt, WeChat, NearbyFeed, etc. It supports discovering the nearby like-minded people, getting familiar with them, and making friends with them.

Just like many current LBS, nearby friend discovery also suffers from location information leakage, which may easily result in the consequence that the LBS users being stalked by other users or by the untrusted LBS servers. Current solutions for protecting the location privacy of the LBS users could be divided into two categories. They are based respectively on location-approximation and k-anonymity. Location-approximation aims at providing quasi-

* Corresponding author. Tel.: +1-716-645-4760 ; fax: +1-716-645-3464.
E-mail address: chencw@buffalo.edu

This research is supported by NSF Grant 0915842.

indistinguishability within a certain area, by approximating the real location of the LBS user[1,2]. For example, if the user is located in the Empire State Building, then from the point of view of the attacker, the user could be anywhere within a certain radius from the Empire State Building. Since the reported location is not the exact location of the user, the service quality will degrade to some extent. Instead of approximating the individual locations, k-anonymity protects the identities of the LBS users by restricting that a user can query for a region such that at least k users of the service are present within that region[3]. This means, it seems to the attacker that the request could come from any LBS users who are around Empire State Building. Another method for achieving k-anonymity is to generate $k-1$ dummy points, and to send k queries to the LBS server[4].

Although both of location-approximation and k-anonymity could preserve users' location privacy for many LBS, neither of them are appropriate for the nearby friend discovery service. First of all, for nearby friend discovery service, identities of the users are necessary in order to find the people in the neighborhood and get familiar with them. Therefore, it is not practical to apply k-anonymity for this service. Second, unlike most other LBS services, such as nearby interest point recommendation and nearby friend alert, nearby friend discovery needs to protect the location privacy of both communicating entities instead of protecting that of only one entity, i.e., the user who is searching for the nearby people, or the user who is being discovered by others. This characteristic will further degrade the service quality when both users are using the approximate locations or $k-1$ dummy points.

By introducing homomorphic cryptography[5], some existing works[6,7] have enabled LBS users with friendship to compute their distances with each other. The end-to-end communications bypass the LBS servers, and hence the users' location information is kept private from the LBS servers. Inspired by these works, we propose to combine the location-approximation technique with the homomorphic cryptography, to protect the privacy of the users in nearby friend discovery service while still providing satisfactory service quality. Specifically, we adopt the location cloaking function proposed by Andres et al.[8] to generate an approximate location with the formal privacy guarantee, i.e. ϵ-geo-indistinguishability, for each user. For this research, we refer the approximate location providing the geo-indistinguishability as *anchor*. As an example, when a user Alice intends to search for the people in her neighborhood, she would generate an anchor and use the anchor to send her friend discovery request to the server. The LBS server produces recommendations only based on the anchors reported by Alice and other users. To guarantee that all users within Alice's interested area can be included in the recommendation list, the LBS server would need to enlarge the recommendation area, resulting in considerable redundancies in the recommendation list. We then propose a two-step refinement procedure, which will firstly make use of the relative positions of the recommended users with respect to Alice's anchor, and then takes advantage of the properties of the homomorphic cryptography, to properly reduce the redundancy and enhance the service quality.

In Section 2, we introduce the related preliminaries, including differential privacy, geo-indistinguishability, and homomorphic cryptography. In Section 3, we present the proposed scheme in more detail. In Section 4, we analyze the security features of the proposed scheme and evaluate its performance. Conclusions are summarized in Section 5.

2. Preliminary

2.1. *Differential Privacy*

Differential privacy[9] is a notion coming from statistical databases. A randomized function $\mathcal{K}$ gives ϵ-differential privacy if for all data sets D_1 and D_2 differing on at most one element, and all $S \subseteq Range(\mathcal{K})$, we have

$$Pr[\mathcal{K}(D_1) \in S] \leq exp(\epsilon) \times Pr[\mathcal{K}(D_2) \in S]$$

A mechanism $\mathcal{K}$ with ϵ-differential privacy is able to guarantee that the outputs of the data set in terms of one request will not become significantly more likely or less likely, even if the participant removes his or her data from the data set. This addresses the concern that any participant might have the leakage of his or her personal information.

2.2. *Geo-Indistinguishability*

Geo-indistinguishability is a variation of differential privacy proposed to provide a formal location privacy protection for the LBS users. The geo-indistinguishability definition is given as follows[8].

A mechanism satisfies ϵ-geo-indistinguishability iff for all observations $S \subseteq \mathbb{Z}$, where x, x' are respectively the real location and the approximate location, and $\mathbb{Z}$ is the range of x and x', we have

$$\frac{P(S|x)}{P(S|x')} \leq e^{\epsilon r} \ (r = d(x, x'))$$

To satisfy ϵ-geo-indistinguishability, the user whose real location is $x \in \mathbb{Z}$ should generate a cloaking location $x' \in \mathbb{Z}$, which can be drawn from the following noise function,

$$D_\varepsilon(r, \theta) = \frac{\varepsilon^2}{2\pi} r e^{-\varepsilon r}$$

In this planar Laplace distribution, r and θ are the distance and the angle with respect to x, respectively. Specifically, the angle θ could be uniformly chosen from $[0, 2\pi)$, and the radius r could be set as $C_\epsilon^{-1}(z)$, where $C_\epsilon(z) = 1 - (1 + \epsilon z)e^{-\epsilon z}$, and z is uniformly chosen from $(0, 1)$.

2.3. Paillier Cryptosystem

Paillier cryptosystem[10] is an asymmetric algorithm for public key cryptography. It is composed of three algorithms as follows.

KeyGenerate: The user randomly selects two large prime numbers p and q with the same length. Then the user computes $n = pq$ and $\lambda = (p - 1)(q - 1)$. Next, the user sets $g = (n + 1)$ and $\mu = (\lambda \bmod n^2)^{-1} \bmod n$. The encryption key is $EK = (n, g)$ and the decryption key is $DK = (\lambda, \mu)$.

Encrypt: The user achieves encryption by choosing a random integer $r \in \mathbb{Z}_n$ and computing the ciphertext $E(m, r) = g^m \cdot r^n \bmod n^2$.

Decrypt: The holder of $DK = (\lambda, \mu)$ recovers the message $m = L((E(m, r))^\lambda \bmod n^2) \cdot \mu \bmod n$, where $L(a) = (a - 1)/n \bmod n$.

Paillier cryptosystem is also an additive homomorphic cryptosystem, with the properties that, for any $m_1, m_2, r_1, r_2 \in \mathbb{Z}_N$, the following two equations hold.

$$E(m_1, r_1) \cdot E(m_2, r_2) = E(m_1 + m_2, r_1 r_2) \bmod N^2$$
$$E(m_1, r_1)^{m_2} = E(m_1 m_2, r_1^{m_2}) \bmod N^2$$

3. The Proposed Scheme

The proposed scheme contains three steps: (1) nearby friend discovery request, (2) friend recommendation, and (3) recommendation list refinement.

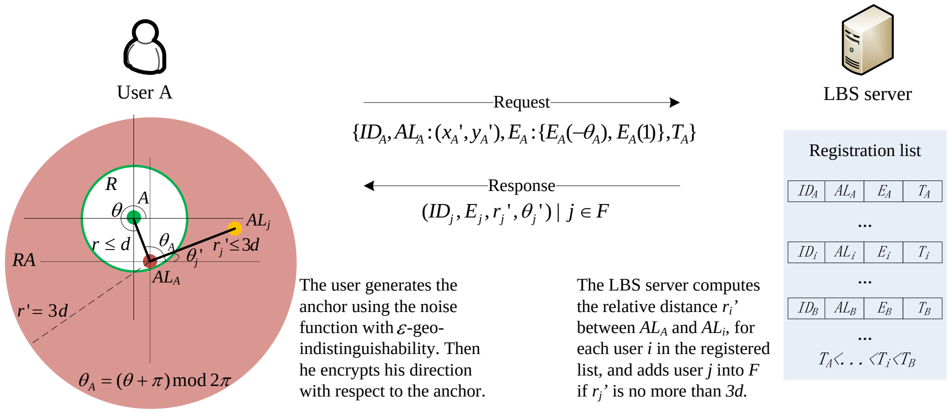
3.1. Nearby Friend Discovery Request

Suppose that a user A, whose real location in the Cartesian reference system is (x_A, y_A), wants to search nearby friends in the neighborhood R, which is a region centered at (x_A, y_A) with the radius of d. To protect the location privacy, A would report his anchor, instead of his real location to the LBS server. The anchor point $AL_A = (x_{A'}, y_{A'})$ could be drawn from the planar Laplace distribution introduced in Section 2.2. That is, $x_{A'} = x_A + rcos\theta, y_{A'} = y_A + rsin\theta$, where $r = \frac{1}{1-(1+\epsilon z)e^{-\epsilon z}} (z \in (0, 1))$, and $\theta \in [0, 2\pi)$. Note that when z approaches 0, r approaches infinity, which means the LBS server would make infinite many recommendations. This is rather unreasonable in the case of nearby friend discovery. Therefore, in this paper, we restrict the range of r, which means, instead of allowing it to grow into infinity, we set it to be no more than d. Hence, we need to uniformly choose $z \in [threshold, 1)$ to generate r. The threshold can be easily obtained given a specific ϵ and d.

Since the LBS server makes recommendations according to the anchors, we let each LBS user compute his or her angle with respect to his or her anchor. For instance, user A's angle is $\theta_A = (\theta + \pi) \bmod (2\pi)$. Then, user A generates the encryption key $EK_A = (n, g)$ and the decryption key $DK_A = (\lambda, \mu)$ in Paillier's cryptosystem. The following cyphertexts are generated using EK_A.

$$E_A = \{E_A(-\theta_A), E_A(1)\}$$

The ciphertexts, along with the user's identity, location of the anchor, and current time stamp, are packed as the request message and forwarded to the LBS server.


User A
LBS server
Request
$\{ID_A, AL_A:(x_A', y_A'), E_A:\{E_A(-\theta_A), E_A(1)\}, T_A\}$
Response
$(ID_j, E_j, r_j', \theta_j') \mid j \in F$
Registration list
ID_A AL_A E_A T_A
...
ID_i AL_i E_i T_i
...
ID_B AL_B E_B T_B
...
$T_A < \ldots < T_i < T_B$
R
θ
A
AL_j
$r \le d$
θ_A
θ_j'
$r_j' \le 3d$
RA
AL_A
$r' = 3d$
$\theta_A = (\theta + \pi) \bmod 2\pi$

The user generates the anchor using the noise function with ε-geo-indistinguishability. Then he encrypts his direction with respect to the anchor.

The LBS server computes the relative distance r_i' between AL_A and AL_i, for each user i in the registered list, and adds user j into F if r_j' is no more than $3d$.

Fig. 1. Nearby friend discovery: requesting and responding

3.2. *Friend Recommendation*

We assume that the LBS server holds a registration list, storing the information of the service users, including the user identities, anchor points, ciphertexts, as well as the time stamps indicating the time when the friend requests were sent. The time stamp is used to restrict the behavior of the LBS users, so that a user in the registration list would be removed when the current time exceeds the corresponding time stamp plus a preset threshold. At the same time, a user is not allowed to send a new request before he or she is removed from the list. This is important to prevent the infinite growth of the registration list, as well as to guarantee the effectiveness of the recommendation, especially in the mobile environment where the users experience high mobility. Besides, time stamp information can also be used to carry out the proposed recommendation procedure, which will be discussed later.

Upon receiving the request from the user A, the LBS server first adds the request message into the registration list. Then, it searches in the list to find the users whose anchors are located within the region RA that is centered at AL_A. For this paper, we assume that the radius of the interested region for each user is the same. Therefore, to guarantee R to be totally included, without the knowledge of user A's real location, the radius of RA should be $3d$. The users located in RA are then added into the recommended friend set F. For each user j in F, the server further computes his or her anchor's position with respect to AL_A in the polar system, i.e., (r_j', θ_j'). This information as well as the identity information and E_j are then forwarded to user A as the recommendation list.

The procedure for the nearby friend discovery requesting and responding is shown in Figure 1. Note that, when there is another user who is in the neighborhood of A sending friend request to the LBS server, user A's information will be forwarded to this user. However, for the users whose time stamp is earlier than that of A, they will not receive user A's information from the LBS server.

3.3. *Recommendation List Refinement*

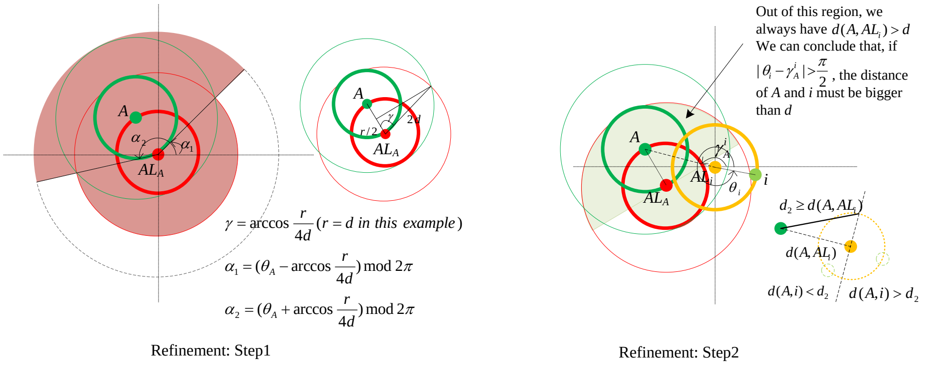
Although the potential nearby friends in region R would be definitely included in F, the service quality is not satisfactory due to the considerable redundancy resulting from enlarged radius of $3d$. In this research, we propose to use the redundancy ratio to measure such a service quality, and adopt the following definition of the redundancy ratio.

$$Redundancy\ Ratio = \frac{number\ of\ the\ redundant\ nodes}{number\ of\ the\ expected\ nodes}$$

If we assume that all of the LBS users are uniformly distributed, then the redundancy ratio could also be defined as

$$Redundancy\ Ratio = \frac{area\ of\ the\ redundant\ region}{area\ of\ the\ expected\ region}$$

We can see that the redundancy ratio of the recommendation set F provided by the LBS server will be as high as eight, since the radius of the recommendation region is three times of that of the expected area. To enhance the service quality, we propose a two-step refinement procedure to reduce the redundancy ratio.


A
α_2
α_1
AL_A
A
$r/2$
γ
$2d$
AL_A
$\gamma = \arccos\frac{r}{4d}$ ($r = d$ in this example)
$\alpha_1 = (\theta_A - \arccos\frac{r}{4d}) \bmod 2\pi$
$\alpha_2 = (\theta_A + \arccos\frac{r}{4d}) \bmod 2\pi$
A
γ_A^i
AL_i
AL_A
θ_i
i
$d_2 \geq d(A, AL_i)$
$d(A, AL_i)$
$d(A,i) < d_2$
$d(A,i) > d_2$

Out of this region, we always have $d(A, AL_i) > d$ We can conclude that, if $|\theta_i - \gamma_A^i| > \frac{\pi}{2}$, the distance of A and i must be bigger than d

Refinement: Step1

Refinement: Step2

Fig. 2. Illustration of the two-step refinement procedure

3.3.1. Step 1: Shrinking

The first step to refine F is to shrink the area of the recommendation region based on the information of $(r_i', \theta_i')|i \in F$. The two parameters used for the shrinking procedure are respectively α_1 and α_2, as shown in Figure 2. Specifically, $\alpha_1 = \theta_A - arccos\frac{r}{4d} \bmod 2\pi$, $\alpha_2 = \theta_A + arccos\frac{r}{4d} \bmod 2\pi$. Then user A shall perform Algorithm 1 as shown below.

Algorithm 1 Shrinking Based Refinement

```
Input: $\alpha_1, \alpha_2, F$, Output: Refined $F$
for All user $i$ in $F$ do
    if ($\theta_i' \in \{(0, min(\alpha_1, \alpha_2)) \bigcup (max(\alpha_1, \alpha_2), 2\pi)\}$ and $r_i' > 2d$) or ($\theta_i' \in (min(\alpha_1, \alpha_2), max(\alpha_1, \alpha_2))$ and $r_i' > 2d + r$) then
        Remove user $i$ from $F$
    end if
end for
return $F$
```

After this step, the area of the region RA would be reduced to RA', with the area of

$$\begin{aligned} S_{RA'} &= \pi(3d)^2 - arccos(\frac{r}{4d}) \cdot ((3d)^2 - (2d+r)^2) - (\pi - arccos(\frac{r}{4d})) \cdot ((3d)^2 - (2d)^2) \\ &= 4\pi d^2 + arccos(\frac{r}{4d}) \cdot (4dr + r^2) \end{aligned}$$

Since the area of the recommended region is reduced, the redundancy ratio will also be reduced, which is equal to

$$Redundancy\ Ratio' = \frac{S_{RA'} - S_R}{S_R} = (3 \sim 5.1)$$

3.3.2. Step 2: Homomorphic Encryption

To further refine F, we propose to make use of homomorphic encryption, based on the Paillier ciphertexts of the users in F. Specifically, user A shall perform Algorithm 2 as shown below.

Note that, only user j who holds the corresponding Paillier decryption key could obtain $\gamma_A^j - \theta_j$ in plaintext. Therefore, to carry out the refinement, user A needs to forward $E_j(\gamma_A^j - \theta_j)$ to user j. Upon receiving the request from user A, user j decrypts the message and compares $|\gamma_A^j - \theta_j|$ with $\frac{\pi}{2}$. If $|\gamma_A^j - \theta_i| > \frac{\pi}{2}$, it means that the distance between user j and user A must be greater than d, since we have $d(A, j) > d(A, AL_i)$ and $d(A, AL_i) > d$, as shown in Figure 2. In this case, user j gives A a negative response. User A then removes user j from F. Otherwise, user j gives A a positive response. User A would keep user j in F. After carrying out the Algorithm 2, the area of the region RA'' where all the users in the output F_1 are located in would become:

$$S_{RA''} = S_{RA'} - (\frac{\pi(d+r)^2}{2} + \frac{\pi d^2}{2}) = 3\pi d^2 - \pi dr - \frac{\pi r^2}{2} + arccos(\frac{r}{4d}) \cdot (4dr + r^2)$$

Algorithm 2 Homomorphic Encryption Based Refinement

```
Input: The output $F$ of Algorithm 1, Output: Refined $F$
$F_1 = F$
for All user $i$ in $F_1$ do
    if $(|\theta_i' - \theta_A| < \frac{\pi}{2}$ and $r_i' \leq d + r)$ or $(|\theta_i' - \theta_A| \geq \frac{\pi}{2}$ and $r_i' \leq d)$ then
        Remove user $i$ from $F_1$
    end if
end for
for All user $j$ in $F_1$ do
    $A$ calculates: $\gamma_A^j$ with respect to $AL_j$, based on the cosine theorem;
    $E_j(1)^{\gamma_A^j} = E_j(\gamma_A^j)$; $E_j(-\theta_j) \cdot E_j(\gamma_A^j) = E_j(\gamma_A^j - \theta_j)$.
    if $|\gamma_A^j - \theta_j| > \frac{\pi}{2}$ then
        Remove user $j$ from $F$
    end if
end for
return $F$
```

In Algorithm 2, we show that if $|\gamma_A^j - \theta_j| > \frac{\pi}{2}$, we will remove j from F. Note that with respect to the anchor, the probability density function (pdf) of the user's real location is also a Laplacian distribution. Hence, the probability of $|\gamma_A^j - \theta_j| > \frac{\pi}{2}$ is $\frac{1}{2}$. The area of RA'' could be accordingly shrunk by half. Then we shall obtain the refreshed redundancy ratio as:

$$Redundancy\ Ratio'' = Redundancy\ Ratio' - \frac{S_{RA''}}{2S_R} = (1.5 \sim 3.3)$$

4. Analysis on the Proposed Scheme

In this section, we evaluate the proposed scheme in terms of its performance in privacy as well as in efficiency.

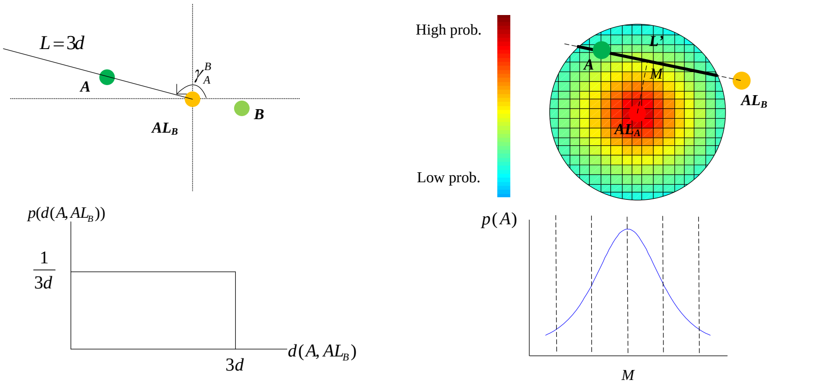
4.1. Privacy Analysis

The proposed scheme contains three steps, i.e., nearby friend discovery requesting, friend recommendation, and friend list refinement. The involved entities include the LBS server, the user who sends the friend request to the LBS server, and the user who responds to other user's refinement request. For the simplicity of presentation, we refer the three entities as user A, server, and user B, respectively.

The entities involved in the first step are user A and the server. Therefore, we consider to protect the location privacy of user A from the LBS server in this step. Since the anchor of user A is generated following the way that satisfies the ϵ-geo-indistinguishability in the region R, the location privacy of user A could be preserved in R. Please refer the proof in [8] for details. Note that it is possible for the server to deliberately store and associate multiple messages sent by user A to figure out his real location if user A sends multiple requests using different anchors generated from the same location. However, this threat could be released in the mobile environment, where users are usually having high mobility, and hence will not always stay in the same location.

In the second step, since the server makes recommendation only based on the anchor information provided by the LBS users, the location privacy of the LBS users in the registration list, including user B, still enjoy the ϵ-geo-indistinguishability in their corresponding regions.

The third step contains two refinement algorithms. First, since shrinking based refinement is only the operation on the side of user A, without involving other entities, the privacy of user A will not be degraded at all. That is, user A still enjoys ϵ-geo-indistinguishability to the attackers. To further refine the recommended friend set F, user A needs to calculate his or her relative angle with respect to the anchor of user B. The angle could be further obtained by user B through Paillier decryption. In this case, user A could be located by user B on a segment L, which starts at the anchor of B with the length of $3d$. As mentioned in Section 3, if B is in the recommendation list of A, B does not have the information about A's anchor. In this case, the probability distribution of A could be treated as the uniform distribution



Case 1: without the knowledge of AL_A, to B, A is uniformly distributed on L at a specific direction with respect to B's anchor

Case 2: with the knowledge of AL_A, the attacker could locate A on the segment L'; On the segment L', the pdf of A's position approaches to linear Laplacian distribution as A approaches to his anchor

Fig. 3. Two cases of the pdf of A's location in 2nd-step refinement

on L. We can show that A still is able to enjoy a ϵ-geo-indistinguishability with a small ϵ in this case. Consider two points X, Y on the segment L, their distance is Δr. Given the observation set S, we have

$$\frac{P(S|Y)}{P(S|X)} = \frac{1/3d}{1/3d} = 1$$

To satisfy the ϵ-geo-indistinguishability, we have

$$\frac{P(S|Y)}{P(S|X)} = e^{\epsilon \Delta r} \Longrightarrow \epsilon = 0$$

Therefore, in this case, user A always enjoys ϵ-geo-indistinguishability on L.

Although we make use of time stamp to restrict the behavior of B in terms of obtaining A's anchor location, B could still send another friend discovery request after he is deleted from the registered list. With the additional information of A's anchor, B could locate A on L', whose length varies from zero to $2d$, depending on the relative position of A and B. Since the anchor is generated by A following the Laplacian distribution in $\mathbb{R}^2$, A's position could also be treated as a Laplacian distribution in $\mathbb{R}^2$ with respect to the anchor AL_A. If we treat the location of AL_A as the origin, then according to [8], the pdf of A's position could be represented as

$$p(A(x_A, y_A)) = \rho(\epsilon)e^{-\epsilon\sqrt{x_A^2+y_A^2}}$$

$\rho(\epsilon)$ is a normalization factor to keep the integral of the pdf function in the region R to be 1. If A could be located on L' by B, without loss of generality, the pdf of A's position could be treated as $p(A(x_A, y_A))$ with a fixed x_0 or y_0, which is, for instance,

$$p'(A(x_A, y_0)) = \rho'(\epsilon, x_A)e^{-\epsilon\sqrt{x_A^2+y_0^2}}$$

$\rho'(\epsilon, x_A)$ is a normalization factor to keep the integral of the pdf function over the whole L' to be 1. With this pdf function, A may not be proved to enjoy ϵ-geo-indistinguishability, except when L' approaches to AL_A, since in this case A's distribution conforms to a linear Laplacian distribution. The two cases in terms of user A's privacy in step-2 refinement are shown in Figure 3.

Now we consider the privacy of B. When receiving the response from B, A could know whether their angle difference with respect to B's anchor exceeds $\frac{\pi}{2}$. This information will determine whether or not the area of the region where B enjoys the ϵ-geo-indistinguishability can be reduced by half. However, since the shape of the pdf of B still conforms to polar Laplacian distribution in the new region, B still satisfies ϵ-geo-indistinguishability in the region with the area of $R/2$. In a recently developed scheme[6], B's privacy will be degraded due to passively responding to the other user's distance request. However, in this scheme, we ensure that B has high chance to enjoy the same degree of privacy in the proposed scheme. On one hand, under the model of the proposed scheme, we can reasonably assume that A will not keep changing the anchor point in order to precisely locate others. Otherwise, A will expose his or her own location to everyone. On the other hand, even A does not care about his or her own location privacy, it is not guaranteed that B will be in A's recommendation list, since neither anchor location nor the time stamp remains unchanged. Formal investigation about this point will be one of our further works.

4.2. Computation Overhead

The main computation cost of the proposed scheme comes from Paillier encryption and decryption, as well as the relative position calculations in polar system. Since Paillier library for smart phone is unavailable currently, we implemented our proposed scheme on laptop HP520 (Intel Core Duo 1.6GHz), which has the comparable computational power as the popular smart phones. Adopting 1024-bit Paillier cryptosystem, and use the 32-bit int type to represent a polar coordinate, the encryption time is 188ms, the decryption time is 374ms, the exponential operation and the relative position calculation is much more lightweight, with the average time of no more than 1ms. In each nearby friend discovery service, for user A, the total cost depends on the number of the users in the output F_1 of Algorithm 2, i.e., N_1, which could be approximated as $(188 \times N_1)ms$. For user B, the total cost would be $374ms$, since the only involved computation is the Paillier decryption. For the LBS server, only relative position calculations are involved. Therefore, the total cost would be $O(M)$, where M is the number of the users in the registration list.

4.3. Communication Overhead

By adopting 1024-bit Paillier cryptosystem, the size of the nearby friend discovery request sent by user A to the server would be $2 \times 1024 + length(ID) + length(timestamp) + 2 \times size(Cartesian\ coordinate)$. The size of the response sent by the server would be $N_2 \times (2 \times 1024 + length(ID) + 2 \times size(polar\ coordinate))$, where N_2 is the number of the users in the unrefined F. In the friend list refinement stage, user A needs to return E_j to each user j in the output F_1 of Algorithm 2, the relative cost would be $2 \times 1024 \times N_1$. Generally, the total communication cost for each nearby friend discovery service is proportional to the number of the users in the neighborhood of a user.

5. Conclusion

We have presented in this paper a strategy that addresses the privacy-aware nearby friend discovery issue. In the proposed scheme, the LBS server makes friend recommendation according to the anchors, which provide geo-indistinguishability for the LBS users within a specific region, instead of the real locations of the users. Then we propose a two-step refinement procedure to remove the redundant recommendations out of the recommendation list, and hence improve the service quality. We show that the LBS users still enjoy ϵ-geo-indistinguishability in a specific region after the refinement. Furthermore, we have also carried out the analysis on the proposed system with regard to both computational overhead and communication overhead.

References

1. S. Peddinti, A. Dsouza, and N. Saxena. Cover Locations: Availing Location-Based Services Without Revealing the Location. WPES '11 Proc. of the 10th annual ACM workshop on Privacy in the electronic society, pages 143-152, 2011.
2. C. Ardagna, M. Cremonini, S. De Capitani di Vimercati, and P. Samarati. An Obfuscation-Based Approach for Protecting Location Privacy. IEEE Transactions on Dependable and Secure Computing, vol.8, no.1, pages 13-27, 2011.
3. M. Mokbel, C. Chow, and W. Aref. The New Casper: Query Processing for Location Services Without Compromising Privacy. VLDB '06 Proc. of the 32nd international conference on Very large data bases, pages 763-774, 2006.
4. P. Shankar, V. Ganapathy, and L. Iftode. Privately Querying Location-Based Services With Sybilquery. Ubicomp '09 Proc. of the 11th international conference on Ubiquitous computing, pages 31-40, 2009.
5. Z. Brakerski, and V. Vaikuntanathan. Efficient Fully Homomorphic Encryption from (Standard) LWE. 2011 IEEE 52nd Annual Symposium on Foundations of Computer Science (FOCS), pages 97-106, October, 2011.
6. X. Li, and T. Jung. Search Me If You Can: Privacy-Preserving Location Query Service. 2013 Proc. IEEE INFOCOM, pages 2760-2768, April 2013.
7. D. Wei, V. Dave, L. Qiu, and Y. Zhang. Secure Friend Discovery in Mobile Social Networks. 2011 Proc. IEEE INFOCOM, pages 1647-1655, April 2011.
8. M. Andres, N. Bordenabe, K. Chatzikokolakis, and C. Palamidessi. Geo-Indistinguishability: Differential Privacy for Location-Based Systems. CCS '13 Proc. of the 2013 ACM SIGSAC Conference on Computer & communications security, pages 901-914, 2013.
9. C. Dwork. Differential Privacy. 33rd International Colloquium on Automata, Languages and Programming, part II (ICALP 2006), vol.4052, pages 112, 2006.
10. P. Paillier. Public-Key Cryptosystems Based on Composite Degree Residuosity Classes. Advances in Cryptology (EUROCRYPT99), vol.1592, pages 223-238, 1999.