World Wide Web (2019) 22:863–883
https://doi.org/10.1007/s11280-018-0550-9

Protecting multi-party privacy in location-aware social point-of-interest recommendation

Weiqi Wang[1] · An Liu[1] · Zhixu Li[1] · Xiangliang Zhang[2] · Qing Li[3] · Xiaofang Zhou[4]

Received: 3 October 2017 / Revised: 13 March 2018 / Accepted: 21 March 2018 /
Published online: 4 April 2018

Abstract Point-of-interest (POI) recommendation has attracted much interest recently because of its significant business potential. Data used in POI recommendation (e.g., user-location check-in matrix) are much more sparse than that used in traditional item (e.g., book and movie) recommendation, which leads to more serious cold start problem. Social POI recommendation has proved to be an effective solution, but most existing works assume that recommenders have access to all required data. This is very rare in practice because these data are generally owned by different entities who are not willing to share their data with others due to privacy and legal concerns. In this paper, we first propose PLAS, a protocol which enables effective POI recommendation without disclosing the sensitive data of every party getting involved in the recommendation. We formally show PLAS is secure in the semi-honest adversary model. To improve its performance. We then adopt the technique of cloaking area by which expensive distance computation over encrypted data is replaced by cheap operation over plaintext. In addition, we utilize the sparsity of check-ins to selectively publish data, thus reducing encryption cost and avoiding unnecessary computation over ciphertext. Experiments on two real datasets show that our protocol is feasible and can scale to large POI recommendation problems in practice.

This article belongs to the Topical Collection: *Special Issue on Deep vs. Shallow: Learning for Emerging Web-scale Data Computing and Applications*
Guest Editors: Jingkuan Song, Shuqiang Jiang, Elisa Ricci, and Zi Huang

✉ An Liu
anliu@suda.edu.cn

[1] School of Computer Science and Technology, Soochow University, Suzhou, China

[2] King Abdullah University of Science and Technology, Thuwal 23955, Saudi Arabia

[3] Department of Computer Science, City University of Hong Kong, Kowloon Tong, Hong Kong

[4] School of Information Technology and Electrical Engineering, The University of Queensland, Brisbane, Australia

Keywords Location privacy · Point-of-interest · Recommendation

1 Introduction

The explosive growth of mobile devices and ubiquitous Internet access results in an emerging location-based service (LBS) market. This kind of services stimulates the accumulation of POI related data. For example, Foursquare announced in 2016 that it has logged more than 100 million POIs worldwide, with more than 600 million photos and 87 million tips, contributed by more than 50 million users. These check-in data reflect users' preferences for places and thus provide a good opportunity for personalized POI recommendation. Like most recommender systems, POI recommendation is significant for both users and LBS providers as it not only helps users to explore new or relevant locations without spending much time on searching but also enables LBS providers to carry out precision marketing for specific POIs [5, 10].

Although developing a personalized POI recommendation can benefit both user's outdoor activities and LBS provider's market competition, it is challenging due to the following reasons. First, a user's check-in decision making process is very complex and could be influenced by many kinds of factors. For example, different friends will have a different influence on a user's check-in behaviors, thus it will be difficult to generate a influence model for each user. Also the geographical distance might have a great impact on user's POI decision. A user often prefers a nearby place to a faraway one. Besides, the extreme sparsity of check-in data makes the cold start problem more serious, which is a big challenge in POI recommendation. Compared with the user-item rating matrix in conventional recommender systems, the user location check-in matrix in POI recommendation is usually much more sparse. For example, the density of Gowalla data set is 2.08×10^{-4}, while the sparsity of Netflix data set is around 0.01.

In the literature, some works propose to solve the cold start problem in POI recommendation using social network [30, 31, 38]. For example, [8] present a solution to simulate the influence of social networks on POI for the better POI recommendation. Meanwhile, some studies such as [19, 39] also take into account geographical influence to assist POI recommendation. A common (implicit) assumption of most existing works is that the recommender owns all of the data, for example, social network and check-ins. It is however rarely the case in practice. Generally, these data are owned by different entities. For example, user check-ins are usually owned by an LBS provider like Foursquare, while social networks are typically owned by a social networking service provider like Facebook. Clearly, a social POI recommender cannot adopt existing techniques directly because it does not always have all these data in hand. This problem will be exacerbated when data owners are reluctant to share their data with the recommender due to privacy concerns.

In order to protect individual privacy and to boost the willingness of each data owner to cooperate with the POI recommender, some studies [15, 18, 20, 25] have been reported in recent years. In [18], Liu et al. focus on the privacy of users' social network, but ingores the privacy of location, which is also important to location owners. [20, 25] both adopt homomorphic encryption to protect the privacy of two sides in the recommendation process. Without considering user's current location, however, the result of POI recommendation will be meaningless sometimes as the recommended POIs may be too far away for users to visit. What's more, homomorphic encryption is very time consuming, so it is difficult to apply these protocols to large recommendation problems .

To overcome the above shortcomings, in this paper, we first utilize homomorphic encryption to design a secure location-aware social recommendation protocol called PLAS, which can protect the sensitive data of every entity getting involved in the process of POI recommendation. Considering homomorphic encryption is very expensive in terms of computation, we also propose two optimizations which can greatly reduce the computation cost of PLAS and make it scalable to large recommendation problems. First, we adopt and adapt cloaking areas to protect users' exact positions, thus avoiding expensive distance calculation over encrypted data. Second, we explore the sparsity of check-ins and propose two methods to selectively publish some useful check-ins, thus avoiding unnecessary computations over encrypted data. The main contributions of our work in this paper can be summarized as follows:

1. We consider privacy-preserving location-aware social POI recommendation, a novel and practical problem setting in which data used for recommendation are distributed among multi-parties who are not willing to share their data in the clear due to privacy concerns.
2. We propose PLAS, a secure protocol for the above problem. PLAS adopts Paillier cryptosystem to encrypt private data and utilizes the homomorphic property of Paillier to make recommendation computable over encrypted data.
3. We design two methods to optimize the performance of PLAS. On one hand, expensive distance computation over encrypted data required in PLAS is replaced by cheap computation over plaintext thanks to the technique of cloaking area. On the other hand, the sparsity of check-in data are explored to design two selective data publication methods which greatly reduces unnecessary publication of check-ins and unnecessary computation on encrypted data.
4. We theoretically analyze the security and complexity of the proposed protocol and empirically evaluate the performance of our protocol with different optimizations on two real-world datasets. Experiments demonstrates our protocol is practical in real applications.

The rest of paper is organized as follows. Section 2 reviews related work and some background knowledge. Section 3 formalizes our problem. In Section 4, we present PLAS, a secure protocol for location-aware social POI recommendation. In Section 5, we propose two methods which can greatly improve the performance of PLAS. Finally, we report the experimental results in Section 6 and conclude our work in Section 7.

2 Related work and preliminary

2.1 Related work

POI recommendation has drawn lots of attention recently in both research community and industry [4, 27]. In [19, 40, 41], they focus on the effect of geographical-influence on POI recommendation and propose different methods to utilize the geographical neighborhood characteristics. In [1, 17], more weight is given to the influence of potential check-ins from friends. Moreover, there has been many studies combing geographical-influence and social-network-influence, such as [14, 42]. With rich data and advanced data processing methods [9, 36], more accurate and personalized POI recommendation can be achieved. However, privacy issues always exist in the process of data collecting, but all the above studies did not take privacy protection into consideration.

Some approaches have been proposed to provide privacy protection for location-based problems. Mix zone [3, 11], k-anonymity [2, 28] and differential privacy [6] are typical methods to guarantee data privacy during location-related queries. However, all of them need a trusted third party to maintain all users' location. In [12], the authors present a solution based on two encryption schemes, Paillier and Rabin, which allows a user to get the exact location to his/her query. In POI recommendation, user location plays an important role, which is a big difference from traditional item recommendation. Lots of approaches aim to solve location-based problems such as fast neighbor searching. In [22, 33–35, 37, 43], some hashing methods are proposed to realize fast neighbor search for a variety of big data. These methods are desirable from both accuracy and efficiency aspects, even for some complicated data such as image and video. However, all the above approaches cannot be applied to our problem as the data in our setting are distributed among multiple parties.

Privacy is also an important problem in other kinds of recommendation. In [7], the authors present a solution for privacy preserving recommendation via homomorphic encryption and data packing. McSherry and Mironov [26] integrate differential privacy into non-social recommender systems. However, their approach will lead to an unacceptable loss of utility when applied to social recommendation. To overcome this weakness, Jorgensen and Yu [16] incorporate a clustering procedure that groups users according to the natural community structure of the social network and significantly reduces the amount of noise. However, all these studies also assume that the data are hold by a single party, which sometimes cannot be met in practice.

2.2 Preliminary

In this paper, we achieve privacy-preserving by using a cryptographic tool called Paillier cryptosystem [29], which are briefly introduced as follows.

Paillier is a public-key cryptosystem whose security is based on an assumption related (but not known to be equivalent) to the hardness of factoring. It consists of the following three algorithms:

- Key generation: Choose two distinct large random primes p, q and compute $N = p * g$. Choose an element $g \in Z^*_{N^2}$. The public key pk is (N, g) and the secret key sk is (p, q).
- Encryption E: Let m be a message in Z_N. It is encrypted by selecting a random number r in Z^*_N and computing
$$c = E(m) = g^m r^N mod\ N^2 \tag{1}$$
where N and g are from the public key pk and c is the ciphertext of m.
- Decryption D: The ciphertext c is decrypted by computing
$$m = D(m) = \frac{(c^\lambda\ mod N^2) - 1}{(g^\lambda\ mod N^2) - 1}\ mod\ N \tag{2}$$
where $\lambda = lcm(p-1, q-1)$ can be computed from the private key sk.

One of the most important properties of Paillier cryptosystem is homomorphic addition. Specifically, multiplying an encryption of m_1 and an encryption of m_2 results in an encryption of $m_1 + m_2$, and raising an encryption of m to a constant k results in an encryption of km, that is,

$$E(m_1)E(m_2) = E(m_1 + m_2) \tag{3}$$
$$E(m)^k = E(km) \tag{4}$$

Besides, Paillier is semantic secure, that is, an adversary cannot learn any partial information about the plaintext from the ciphertext. As a result, it is also a probabilistic encryption scheme, which means when encrypting the same message several times, it will produce different ciphertexts. This is clear from (1) where a random number r is used in encryption.

3 Problem definition

3.1 System model

Figure 1 shows the system model of our location-aware social POI recommendation where the recommender R wants to recommend a set of POIs to a user u. The recommendation algorithm adopted by R is a variation of the well-known collaborative filtering (CF) that has been widely used recently by commercial recommender systems (e.g., Amazon and Netflix). More specifically, the conventional CF is enhanced as follows. To solve the cold start problem, the similarity between two users is measured by their social relations, instead of their check-in histories. Besides, the distance between a user's current location and the recommended POI is considered in the recommendation, as in practice users are often not willing to visit POIs that are far away. Formally, the score of recommending a POI p to a user u is calculated as follows:

$$s_{u,p} = \sum_{v \in U, v \neq u} \omega_{u,v} \times c_{v,p} \times f_{u,p} \tag{5}$$

where $\omega_{u,v}$ is the similarity between u and v based on their social relations, $c_{v,p}$ is the number of check-in that v has ever done at p, and $f_{u,p}$ is the geographical-influence between u's current location and p. To model geographical-influence, we assume that every user has

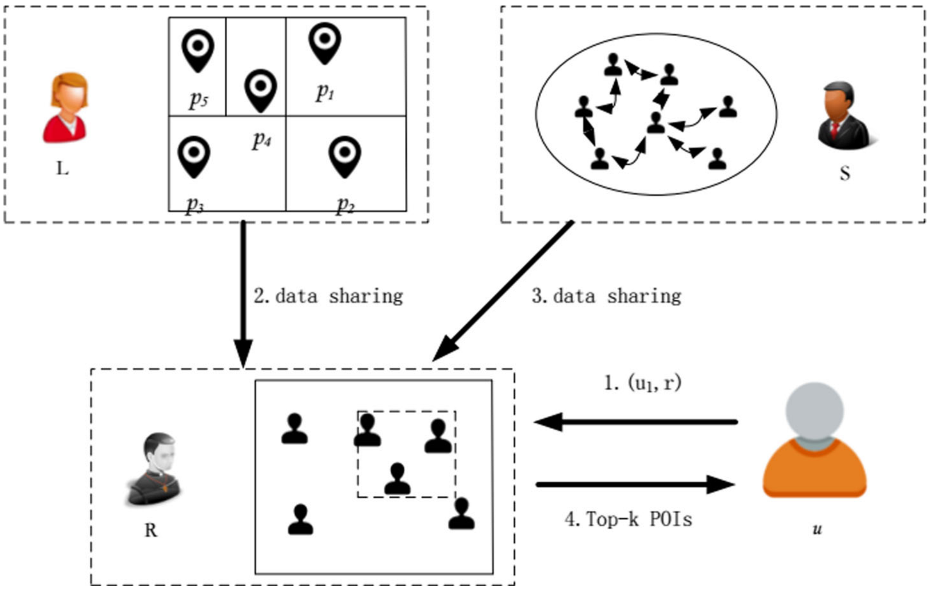


Figure 1 System model

a willing range r, i.e., the maximum distance that he/she is willing to travel. Then $f_{u,p}$ is calculated as follows:

$$f_{u,p} = \begin{cases} 1 - \frac{d_{u,p}}{r} & , d_{u,p} \leq r \\ 0 & , d_{u,p} > r \end{cases} \tag{6}$$

where $d_{u,p}$ is the distance between u and p. When $d_{u,p}$ is larger than r, $f_{u,p}$ is set to 0, so the score $s_{u,p}$ equals to 0, indicating that it is meaningless to recommend p to u as it is beyond the range that u is willing to travel.

Once all scores are evaluated, a set of POIs with the highest scores are recommended to the user u. It is significant to examine the data required in this recommendation procedure. From (5) and (6), there are three types of data: check-in data, social data, and geographical data of user u. In our problem setting, these data belong to different entities and they are defined as follows.

Definition 1 (Check-in data). The check-in data hold by a location-based service (LBS) provider L is a $|U| \times |P|$ matrix C where U is a set of users and P is a set of POIs. Each entry $c_{u,p} \in C$ is the number of check-ins that user $u \in U$ has ever done at the location $p \in P$, indicating u has not visited p if $c_{u,p} = 0$ and otherwise the check-in frequency u has on p.

Definition 2 (Social data). The social data hold by a social networking service (SNS) provider S is an undirected graph $G = (U, E)$ where U is a set of users, and $E \subseteq U \times U$ is a set of edges. An edge $(u, v) \in E$ indicates that two users $u, v \in U$ has some kind of social relation and the weight $\omega_{u,v}$ assigned to this edge represents the social similarity between them.

Definition 3 (Geographical data of user u). The geographical data hold by a user u is his/her current position l_u and willing range r. This data can be also accessed by the recommender R as a prerequisite for using the recommendation service provided by R.

It is clear from the above definitions that the data used in the recommendation belong to different parties. These data are not easy to collect and maintain, and sometimes are even regarded as trade secrets, so it is unrealistic to assume that L and S are willing to share their data with R. Consequently, we have to face a challenging problem, privacy-preserving location-aware social POI recommendation. Informally, we need to make recommendations while keeping the data of each parties secret. To formally define this problem, we first introduce the adversary model and security definition in the next subsection.

3.2 Adversary model and security definition

In this paper, we consider a typical adversary model, that is, the semi-honest model [13], that has been widely accepted in a variety of privacy-preserving problem domains [21, 23, 24]. Specifically, all parties in this model are assumed to be semi-honest, that is, they follow the recommendation protocol exactly as specified, but may try to learn as much as possible about other parties' private input from what they see during the protocol's execution. We also assume that no two of them collude. As stated in [32], the assumption of non-collusion between two well-established companies (real examples for R, L, and S are Google Place, Foursquare, and Facebook, respectively) is reasonable as the collusion will damage their reputation and consequently reduce their revenues.

Further we adopt the well-known real-ideal paradigm [13] to define the security of a protocol. Intuitively, a protocol is secure or privacy-preserving if every party involved in the protocol learns no more knowledge from the execution of the protocol than the knowledge that this party is entitled to know. This can be formally defined by the real-ideal paradigm as follows: for all adversaries, there exists a probabilistic polynomial-time simulator, so that the view of the adversary in the real world and the view of the simulator in the ideal world are computationally indistinguishable.

Let $\mathcal{F}(x_1, \cdots, x_n) = (\mathcal{F}_1, \cdots, \mathcal{F}_n)$ be an n-ary functionality, where x_i and $\mathcal{F}_i$ are the i-th party P_i's input and output, respectively. For $I = \{i_1, \cdots, i_t\} \subseteq [n] \overset{def}{=} \{1, \cdots, n\}$, let $\mathcal{F}_I$ denote the subsequence $\mathcal{F}_{i_1}, \cdots, \mathcal{F}_{i_t}$. Let Π be a n-party protocol for computing $\mathcal{F}$. The view of P_i during an execution of Π on $\overline{x} = \{x_1, \cdots, x_n\}$, denoted as $view_i^{\Pi}(\overline{x})$, is (x_i, r, m_i) where r represents the outcome of P_i's internal coin tosses and m_i represents the messages that it has received. For $I = \{i_1, \cdots, i_t\}$, let $view_I^{\Pi}(\overline{x}) \overset{def}{=} (I, view_{i_1}^{\Pi}(\overline{x}), \cdots, view_{i_t}^{\Pi}(\overline{x}))$. The security of Π is formally defined as follows:

Definition 4 (**Secure protocol under semi-honest model** [13]) A protocol Π privately computes $\mathcal{F}$ if there exists a probabilistic polynomial-time simulator S, such that for every $I \subseteq [n]$, it holds that:

$$S(I, (x_{i_1}, \cdots, x_{i_t}), \mathcal{F}_I(\overline{x})) \equiv view_I^{\Pi}(\overline{x}) \tag{7}$$

where $\equiv$ denotes computational indistinguishability.

The above equation asserts that the view of every party in I can be efficiently simulated based on its input and output. That is, it cannot derive extra information during an execution of the protocol Π, indicating Π is secure or privacy-preserving.

3.3 Problem statement

We first review the required private data when making recommendation for a given user u. For R, its private data are u's location l_u and willing range r. For S, its private data are social similarity $\omega_{u,v}$ between u and any other user v. For L, its private data are the number of check-ins $c_{v,l}$ for any user v. Based on this observation and the security definition presented in last subsection, we formally define the problem of privacy-preserving location-aware social POI recommendation.

Definition 5 (Privacy-preserving location-aware social POI recommendation) Given the private check-in data hold by L, the private social data hold by S, and the private geographical data of user u hold by R, find a k-size ordered POI set $L_k = \{l_{i_1}, l_{i_2}, ..., l_{i_k}\}$ for u, such that:

1. $d_{u,l} \leq r$ for all $l \in L_k$,
2. $s_{u,l_i} \geq s_{u,l_j}$ for $l_i \in L_k$ and $l_j \in L \setminus L_k$,
3. $s_{u,l_1} \geq s_{u,l_2} \geq ... \geq s_{u,l_{i_k}}$,
4. equation (7) holds.

In the above definition, the first requirement is about location-aware, that is, the recommended POIs must be in u's willing range with respect to his/her current location. The second requirement indicates that the recommended L_k should be top-k POIs while the third says that the recommendation result should be ordered to facilitate u's decision making about where to go. The last requirement is about security, that is, the private data of every party should be protected during the procedure of POI recommendation.

4 Secure protocol for location-aware social POI recommendation

In this section, we present a fully secure protocol PLAS for location-aware social POI recommendation. We describe the details of PLAS in Section 4.1, prove its security in Section 4.2, and finally analyze its complexity in Section 4.3. Table 1 summarizes the major notations used in this paper.

4.1 Protocol

PLAS adopts Paillier cryptosystem to protect the private data of every party getting involved in the recommendation. The principle idea is to encrypt private data and then utilize the homomorphic property of Paillier to make the whole recommendation algorithm computable over encrypted data. In particular, using the homomorphic addition property shown in (3) and (4), (5) can be rewritten over encrypted data as follows:

$$E(s_{u,p}) = (\prod_{v}(E(c_{v,p}))^{\omega_{u,v}})^{f_{u,p}} \tag{8}$$

Following the above equation, Figure 2 describes from a high level view how PLAS works and Algorithm 1 shows its detailed procedure. We explain it in details as follows.. At first, R encrypts u's location $l_u = (x_u, y_u)$ using S's pulibc key and sends three ciphertexts $E_S(x_u^2 + y_u^2)$, $E_S(x_u)$, and $E_S(y_u)$ to L. Upon receiving these data, L calculates the square of $d_{u,p}$ in the encrypted form as follows:

$$E_S(d_{u,p}^2) = E_S(x_u^2 + y_u^2)E_S(x_u)^{-2x_p}E_S(y_u)^{-2y_p}E_S(x_p^2 + y_p^2) \tag{9}$$

by noting that $d_{u,p}^2 = (x_u - x_p)^2 + (y_u - y_p)^2$ holds over the plaintext. This encrypted square distance is then masked by R using a random noise m, as shown in the following equation:

$$E_S(\bar{d}_{u,p}^2) = E_S(d_{u,p}^2)E_S(m) \tag{10}$$

Table 1 Summary of notation

Notation	Meaning
R	POI recommender
L	location-based service provider
S	social networking service provider
$\omega_{u,v}$	social similarity between u and v
$c_{u,p}$	number of check-ins that user u has ever done at p
$f_{u,p}$	geographical-influence between u and p
$s_{u,p}$	score of recommending p to u
$\bar{s}_{u,p}$	masked score of recommending p to u
$E_L(x)$	encryption of message x using L's public key
$E_S(x)$	encryption of message x using S's public key

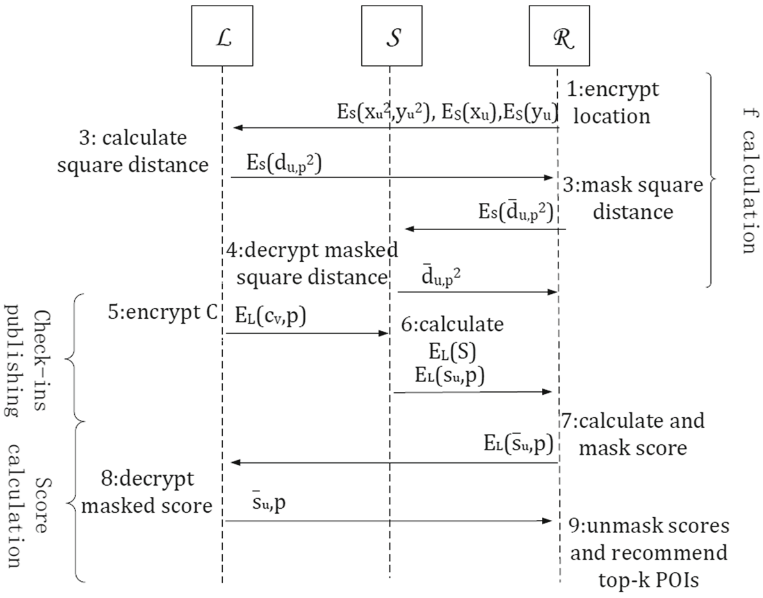


Figure 2 Overview of Paillier protocol

The purpose of masking is to ensure there is no information leakage during later decryption. Now, R sends the masked square distance to S for decryption. With the secret key, S is able to obtain the masked square distance in the clear. After that, R removes the mask by subtracting the added noise and calculates $f_{u,p}$ according to (6).

Next, to calculate scores $s_{u,p}$, L encrypts its private data C using its own public key and sends the encrypted C to S. For every POI p, S calculates as follows a weighted sum of number of check-ins where the weights are the social similarities it has:

$$E_L(s_{u,p}) = \prod_{v \in U \setminus \{u\}} (E_L(c_{v,p}))^{\omega_{u,v}} \tag{11}$$

Based on this intermediate result, R only needs to do some simple exponentiation to obtain the final value of $s_{u,p}$ (in the encrypted form), as long as it knows $f_{u,p}$ in the clear, and $f_{u,p}$ can be calculated as discussed above.

Holding encrypted score $E_L(s_{u,p})$ for every p, R's next job is to find top-k POIs with the highest scores. This can be done similarly using the mask-decrypt-unmask idea. Specifically, R masks $E_L(s_{u,p})$ for every p as follows:

$$E_L(\bar{s}_{u,p}) = E_L(s_{u,p})E_L(m) \tag{12}$$

where m is a random noise chosen by R. By decrypting $E_L(\bar{s}_{u,p})$, L obtains $\bar{s}_{u,p}$, a masked score that recommending p to u. By unmasking these scores, it is easy for R to find top-k POIs for u.

Algorithm 1 PLAS: A protocol for privacy-preserving location-aware social POI recommendation

```
Input: $u$'s location hold by $R$, check-ins hold by $L$, and social similarity hold by $S$
Output: top-$k$ POIs
1: Key generation: $L$ and $S$ generate their Paillier key pairs and distribute their public
   keys to others
2: $R$ encrypts $x_u^2 + y_u^2$ , $x_u$, and $y_u$ by $S$'s public key and sends the results to $L$
3: for each POI $p_i$ $(1 \leq i \leq n)$ do
4:     $L$ encrypts $x_i^2 + y_i^2$ by $pk_S$
5:     $L$ computes $E_S(d(l_u, l_i)^2) = E_S(x_u^2 + y_u^2)E_S(x_u)^{-2x_i}E_S(y_u)^{-2y_i}E_S(x_i^2 + y_i^2)$
6: end for
7: $L$ sends all $E_S(d(u, p)^2)$ to $R$
8: $R$ masks square distance as $E_S(\bar{d}_{u,p}^2) = E_S(d_{u,p}^2)E_S(m)$ and sends it to $S$
9: $S$ decrypts masked square distance to $\bar{d}_{u,p}^2$ and returns it to $R$
10: $R$ calculates corresponding $f_{u,p}$ between POIs and user $u$
11: $L$ encrypts every check-in as $E_L c_{v,p}$ and sends it to $S$
12: $S$ calculates $E_L(s_{u,p}) = \prod_{v \in U \setminus \{u\}}(E_L(c_{v,p}))^{\omega_{u,v}}$
13: $S$ sends all POI scores $E_L(s_{u,p})$ to $R$
14: $R$ calculates final scores $E_L(S)$
15: $R$ masks $E_L(S)$ to $E_L(S')$ and sends $E_L(S')$ to $L$
16: $L$ decrypts $E_L(S')$ and sends $S'$ to $R$
17: $R$ unmasks $S'$ and finds top-$k$ POIs
```

4.2 Security analysis

In this section, we show the security of PLAS. Informally, PLAS is secure due to two reasons: 1) the parties who do not have the secret key cannot deduce any information from the encrypted data they received, which is guaranteed by the security of Paillier cryptosystem; 2) the data received by the parties who have the secret key are masked by random noise added by other parties, so no information leakage would occur considering that no two parties are allowed to collude as assumed by the system model.

We give below a formal security analysis based on simulator. Note that we only need to show the view of every party in PLAS can be efficiently simulated based on its input and output only.

We first consider L. During an execution of PLAS, the messages received by L can be divided into two groups: one is three ciphertexts about u's location and the other is $|P|$ ciphertexts about the masked scores. As the former is encrypted by S's public key and the latter by L's public key, its view is $view_L = \{E_S(x_u^2 + y_u^2), E_S(x_u), E_S(y_u), \bar{s}_{u,p}|p \in P\}$. To construct a probabilistic polynomial-time simulator $\mathcal{S}_L$ that can simulate L's view, we let $\mathcal{S}_L$ generate $|P| + 3$ random numbers uniformly distributed in Z_N. It is easy to verify that these numbers are computationally indistinguishable from $view_L$ due to the semantic security of Paillier cryptosystem and the randomness of $\bar{s}_{u,p}$ caused by random mask chosen by R.

Next we show that there is a probabilistic polynomial-time simulator $\mathcal{S}_S$ which can simulate S's view efficiently. Similar to the case of L, what S receives during an execution of PLAS is $|U||P|$ encrypted check-ins in the form of $E_L(c_{u,p})$ and $|P|$ encrypted

Table 2 Computation cost of PLAS

	enc.	dec.	mul.	exp.
R	$\lvert P\rvert$	0	$\lvert P\rvert + \lvert P\rvert$	$\lvert P\rvert$
S	0	$\lvert P\rvert$	$\lvert P\rvert\lvert U\rvert$	$\lvert P\rvert\lvert U\rvert$
L	$\lvert P\rvert + \lvert P\rvert\lvert U\rvert$	$\lvert P\rvert$	$3\lvert P\rvert$	$2\lvert P\rvert$

masked square distances in the form of $\bar{d}^2_{u,p}$. Clearly, in this case, we simply let $\mathcal{S}_S$ generate $|U||P| + |P|$ random numbers uniformly distributed in Z_N to simulate $views_S$.

The case of R is a little bit complicated as it can learn some intermediate result, denoted as K, during an execution of PLAS. The proof presented below first shows it is possible to construct a probabilistic polynomial-time simulator $\mathcal{S}_R$ to simulate R's view efficiently based on K, and then demonstrates the probability that R learns the private data of L or S from K is negligible. Based on the messages received by R, it is clear that $view_R = \{E_L(s_{u,p})|p \in P\} \bigcup \{d^2_{u,p}|p \in P\} \bigcup \{s_{u,p}|p \in P\}$. To simulate it, we let $\mathcal{S}_R$ generate $|P|$ random numbers uniformly distributed in Z_N and a $2|P|$-size set of fixed numbers, that is, $K = \{d^2_{u,p}, s_{u,p}|p \in P\}$. It is easy to see that these views are computationally indistinguishable since Paillier are semantic secure. Now consider what L can learn from K. For every $d^2_{u,p}$, R can construct a equation $(x_p - x_u)^2 + (y_p - y_u)^2 = d^2_{u,p}$. There are two variables x_p and y_p in this equation, and clearly R can learn the location of p if the equation has a unique solution. For positive $d_{u,p}$, however, this equation has infinitely many solutions, so R cannot learn the private data of L. The case of $s_{u,p}$ is similar, so the proof is omitted here.

Based on the above discussion, we can conclude that PLAS is secure in the semi-honest model.

4.3 Complexity analysis

Table 2 shows the computation cost of PLAS in terms of the number of encryptions, decryptions, multiplications (over ciphertext), and exponentiations (over ciphertext) performed by different parties. We ignore common operations on plaintext as their cost are negligible compared with the aforementioned four operations on ciphertext. Clearly, L has the most computation complexity among all parties. In step 1, L needs to encrypt $|P||U|$ check-ins and in step 4 it needs to execute $|P|$ encryption, $3|P|$ multiplications, $2|P|$ exponentiations to calculate the square distance. Moreover, in step 8, L needs to decrypt $|P|$ masked scores. For S, it needs to execute $|P||U|$ multiplications and exponentiations to calculate scores in step 2, and decrypts $|P|$ masked square distance ciphertexts in step 6. In steps 5 and 7, R needs to execute $|P|$ multiplications and encryptions to mask square distance and score.

Table 3 shows the computation cost of every party during the execution of PLAS. The communication cost of PLAS largely comes from the ciphertexts transferred between different parties. For L, it needs to receive total $4|P|$ ciphertexts in step 4 and step 8. In addition, it needs to send $|P||U|$ ciphertexts in step 1 and $|P|$ ciphertexts in step 4. For S, it receives

Table 3 Communication cost of PLAS

	L	S	R
PLAS	$\lvert P\rvert(\lvert U\rvert + 5)$	$\lvert P\rvert(\lvert U\rvert + 2)$	$4\lvert P\rvert$

$|P||U|$ ciphertexts and sends $|P|$ ciphertexts in step 2 and receives $|P|$ ciphertexts in step 6. For R, it receives $|P|$ ciphertexts in step 3 and step 5 and sends $|P|$ ciphertexts in step 5 and step 7.

5 Optimization

Though PLAS can protect the private data of every party got involved in the procedure of recommendation, it does not scale well to large problems due to its high computational complexity. From the discussion presented in Section 4.3, we can see that the heavy computational cost mainly comes from two tasks in PLAS: geographical-influence estimation and check-ins publication. In the former task, the distance between u and every POI $p \in |P|$ needs to be calculated over encrypted data. In the latter task, $|P||U|$ check-ins need to be encrypted to enable subsequent computation. In this section we propose two methods to reduce these expensive operations at the cost of sacrificing some security, resulting in weaker secure protocols. This strategy, that is, a trade-off between security and efficiency, is common in practice. In Section 5.1, geographical-influence is estimated over cloaking area in the clear rather than ciphertexts generated by Paillier. In Section 5.2, the sparsity of check-in matrix is explored and only a small portion of check-ins are encrypted and published.

5.1 Geographical-influence estimation based on cloaking area

In PLAS, u's location is fully protected by Paillier encryption. To reduce computational cost, we use cloaking area, a technique proposed in [28], to protect u's location. The principle idea is that instead of giving u's exact location to L, R sends L an area g such that: 1) u is in g; and 2) g can meet some security requirements. Based on g, L can estimate the approximate distance between u and a POI p without learning u's exact location. Here, the security of g is judged from two aspects: 1) there are at least k users in g; and 2) g's area is larger than a threshold A. The first property ensures that u is indistinguishable from other $k-1$ users in the same area even the adversary has some background knowledge, for example, he/she knows a user must be in a location in g. Note that when $k = 1$, the adversary can immediately learn u's location. The second property ensures that the adversary cannot learn u's location when g is very small but crowded (that is, there are more than k users in g). Note that, the security of cloaking area depends on the values of k and A, and they should be set appropriately according to specific applications. To guarantee such an area always exists, k and A should be less than the total users registered in R and the total spatial area, respectively.

Algorithm 2 shows how to find a cloaking area for a user. As a prerequisite, R divides the entire area into a grid with a set of cells. For each cell g, R maintains a list T recording users who are currently in g and an indicator I whose initial value is 0 and will be set to 1 once it is visited. The algorithm checks in lines 1-2 whether the current area g satisfies the specified security requirements and return g if yes. Otherwise, g is expanded by adding a neighbor cell g_N that has not been visited before. When g has more than one new neighbor, the one with most users is selected if the number of users in g is less than k (line 5), as we want to keep g as small as possible to avoid unnecessary computation later, or the one nearest to u is selected when g's area is less than A (line 7), as we want to make g's center as near to u as possible to improve the accuracy of geographical-influence estimation. Lines 10-19 address the case when g's area cannot meet security requirement and line 21 deal with the case when a larger g containing more users is needed. Given a user u, this algorithm takes g_u, k, A as input where g_u is the cell where u is located in.

Algorithm 2 getArea(g, k, A): Construct a cloaking area for a user

```
Input: Area $g$, security parameters $k$ and $A$
Output: A cloaking area $g$ satisfies security requirement
1: if $|g.T| \geq k$ and $g.Area \geq A$ then
2:     return $g$
3: end if
4: if $|g.T| < k$ then
5:     $g_N \leftarrow$ A new neighbor cell of $g$ which has the most users
6: else
7:     $g_N \leftarrow$ A new neighbor cell of $g$ which is nearest to $u$
8: end if
9: $g \leftarrow g \bigcup g_N$
10: if $|g.T| \geq k$ then
11:     if $g.Area \geq A$ then
12:         return $g$
13:     else
14:         while $g.Area < A$ do
15:             $g_N \leftarrow$ A new neighbor cell of $g$ which is nearest to $u$
16:             $g \leftarrow g \bigcup g_N$
17:         end while
18:         return $g$
19:     end if
20: else
21:     return getArea($g, k, A$)
22: end if
```

Once the cloaking area g for a user u has been figured out, it is sent to L for geographical-influence estimation. In particular, L uses the distance between g_c and p to approximate the distance between u and p. Next, L sends all approximated distances to R for subsequent geographical-influence estimation. A special case needs to be noted where the actual distance $d_{u,p}$ is less than u's willing range r but the approximated distance $d_{g_c,p}$ is larger than r. If (6) is still used in this case, then p will be filtered out, making the recommendation result inaccurate. To address this problem, (6) is modified as follows:

$$f_{u,p} = \begin{cases} 1 - \frac{d_{g_c,p}}{r+\max\{d(g_c,g_e)\}} & , d_{g_c,p} \leq r + \max\{d(g_c, g_e)\} \\ 0 & , d_{g_c,p} > r + \max\{d(g_c, g_e)\} \end{cases} \tag{13}$$

where $\max\{d(g_c, g_e)\}$ is the maximum distance between the center of g to its edges. The new equation ensures that POIs within u's willing range are still under consideration when u's location is hidden in a cloaking area.

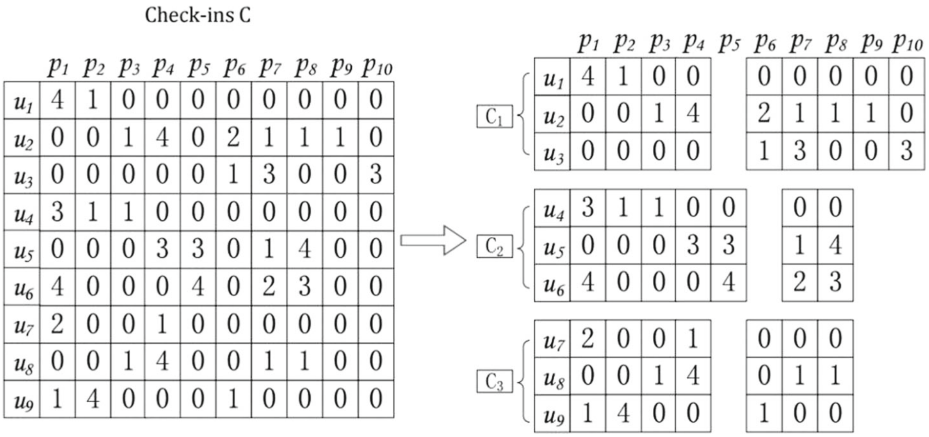
Using cloaking area, the estimation of geographical-influence is carried out completely over plaintext, so its computation cost is negligible compared with the cost of encryption and decryption required in PLAS. In terms of security, the adversary now can learn the area where u is located in, but the property that the cloaking area has still provides flexible security for private data.

5.2 Selective publication of check-ins

Recall that all $|U||P|$ elements in the check-in matrix are encrypted and published in PLAS, which incurs prohibitive cost in the process of data publication as well as subsequent score computation. However, encrypting all check-ins is not necessary, for example, only POIs within u's willing range need to be published to S, as it is meaningless to compute the score of POIs beyond u's willing range. Even after some POIs are filtering out based on u's willing range, however, the remaining check-in matrix is still large yet sparse. To further improve computation efficiency, the sparsity of check-in matrix C can be utilized since the zeros in C does not contribute to the score computation. A straightforward way is then to only publish non-zero check-ins in C, and this can avoid all unnecessary computation over encrypted data. This method, however, will let the adversary learn additional information, that is, some users must visited some POIs.

Following the idea of k-anonymity, we propose a method called KAP which selectively publishes some check-ins to deal with the above problem. Specifically, the matrix C is first divided into $\lceil \frac{|U|}{k} \rceil$ sub matrices such that each matrix C_i has k rows (the last matrix can be padded to ensure k rows). For each C_i, if a column is all zeros, that is, the corresponding POI has not been visited by any of these k users, it is deleted from the matrix to avoid necessary check-ins publication. Note that such a deletion is safe from the security point of view as knowing that a user does not visit some POIs does not contribute to the success of inferring which POIs the user has visited and how often. Finally, these $\lceil \frac{|U|}{k} \rceil$ matrices are encrypted by Paillier for publication. Figure 3 illustrates how KAP works on a 9×10 matrix when $k = 3$. In C_2, p_6 has not been visited by any user, so the column of p_6 is deleted. Using KAP, 18 elements are removed from C, resulting in 20% decrease in the number of encryption. By grouping k users together and deleting only the column with k zeros, the adversary cannot conclude whether or not a user has visited a POI according to the encrypted check-in matrix. On the other hand, these k users are indistinguishable due to the same POI set and the semantic security of Paillier cryptosystem.

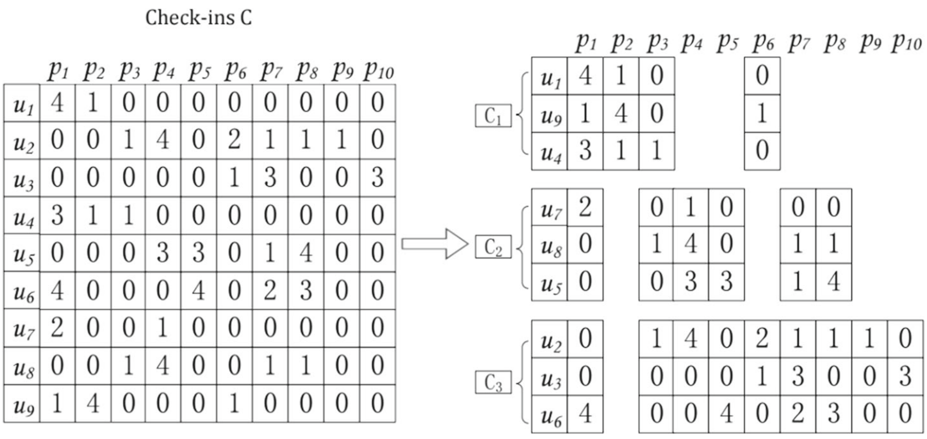
The sub matrices returned by KAP are still sparse sometimes as users are grouped together at random. To further improve the performance of selective publication, we propose another publication method called SKAP in which users are sorted before grouping so



	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
u_1	4	1	0	0	0	0	0	0	0	0
u_2	0	0	1	4	0	2	1	1	1	0
u_3	0	0	0	0	0	1	3	0	0	3
u_4	3	1	1	0	0	0	0	0	0	0
u_5	0	0	0	3	3	0	1	4	0	0
u_6	4	0	0	0	4	0	2	3	0	0
u_7	2	0	0	1	0	0	0	0	0	0
u_8	0	0	1	4	0	0	1	1	0	0
u_9	1	4	0	0	0	1	0	0	0	0

		p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
C_1	u_1	4	1	0	0		0	0	0	0	0
	u_2	0	0	1	4		2	1	1	1	0
	u_3	0	0	0	0		1	3	0	0	3
C_2	u_4	3	1	1	0	0		0	0		
	u_5	0	0	0	3	3		1	4		
	u_6	4	0	0	0	4		2	3		
C_3	u_7	2	0	0	1		0	0	0		
	u_8	0	0	1	4		0	1	1		
	u_9	1	4	0	0		1	0	0		

Figure 3 k-anonymity publication of check-ins


Check-ins C
C_1
C_2
C_3

	p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
u_1	4	1	0	0	0	0	0	0	0	0
u_2	0	0	1	4	0	2	1	1	1	0
u_3	0	0	0	0	0	1	3	0	0	3
u_4	3	1	1	0	0	0	0	0	0	0
u_5	0	0	0	3	3	0	1	4	0	0
u_6	4	0	0	0	4	0	2	3	0	0
u_7	2	0	0	1	0	0	0	0	0	0
u_8	0	0	1	4	0	0	1	1	0	0
u_9	1	4	0	0	0	1	0	0	0	0

		p_1	p_2	p_3	p_4	p_5	p_6	p_7	p_8	p_9	p_{10}
C_1	u_1	4	1	0			0				
	u_9	1	4	0			1				
	u_4	3	1	1			0				
C_2	u_7	2		0	1	0		0	0		
	u_8	0		1	4	0		1	1		
	u_5	0		0	3	3		1	4		
C_3	u_2	0		1	4	0	2	1	1	1	0
	u_3	0		0	0	0	1	3	0	0	3
	u_6	4		0	0	4	0	2	3	0	0

Figure 4 *k*-anonymity publication of sorted check-ins

that more columns with all zeros may exist in the sub matrices. Here we adopt Hamming distance as the metric for sorting. Informally, the Hamming distance between two strings of equal length is the number of positions at which the corresponding symbols are different. At first, a row of C, that is, a user's check-in at $|P|$ POIs, is mapped to a $|P|$-bits string where 1 represents a non-zero check-in and 0 otherwise. Then, all rows in C are sorted based on the Hamming distance between their corresponding bit strings. The remaining procedure is the same as KAP and is omitted here. Figure 4 depicts how SKAP works on the same matrix shown in Figure 3 when $k = 3$. Now 33 elements are removed from C, which is better than KAP.

6 Performance evaluation

6.1 Experimental setting

To the best of our knowledge, this is the first piece of work towards privacy-preserving location-aware social POI recommendation. Hence we only evaluate in this paper the performance of PLAS and the proposed optimizations. In particular, we consider three variations of PLAS: 1) PLAS-CA in which the geographical-influence estimation in PLAS is optimized by cloaking area; 2) PLAS-CA-KAP in which the geographical-influence estimation in PLAS is optimized by cloaking area and the check-ins publication in PLAS is performed by KAP; and 3) PLAS-CA-SKAP in which the geographical-influence estimation in PLAS is optimized by cloaking area and the check-ins publication in PLAS is performed by SKAP. It should be noted that we do not evaluate PLAS without any optimization as the data sets we used are too big to make PLAS completed in a reasonable time. This is just the reason we need optimization. These three protocols are evaluated on two real datasets: Gowalla[1] and Yelp.[2] The check-in matrix in Gowalla contains 151,075 POIs and 6,160 users, and in Yelp contains 15,583 POIs and 70,817 users.

[1]https://snap.stanford.edu/data/loc-gowalla.html
[2]https://www.yelp.com/datasetchallenge

Several parameters are varied in experiments to show the dynamic performance of different protocols. In particular, a user's willing range r varies from 1km to 6km, and 4km is the default value unless specified otherwise. The parameter k required in the selective publication of check-ins belongs to $k = \{20, 40, 60, 80, 100\}$ and 40 is the default value unless specified otherwise. For security parameter of Paillier, we refer to NIST recommendations (2016)[3] and set the key length to be 1024 as it is appropriate for current applications. All experiments are performed on a server with 40 2.4GHz CPUs, 64GB RAM, JDK 7 and Linux version 3.10.

As our work focuses on efficiency, we consider two metrics: CPU time and communication cost. The CPU time of a protocol is defined to be the sum of computation of every party involved in the protocol. It is reasonable here because the computation tasks performed by different parties in all the protocols are actually in a sequential order. Moreover, we only consider on-line CPU time, which means that the CPU time of all computation tasks that can be done off-line are not counted here. Moreover, we ignore the CPU time of plaintext calculation, such as finding top-k scores in the clear and computing cloaking area, because it is negligible compared to the CPU time of ciphertext calculation. The communication cost of a protocol is defined to be the amount of data that need to be transferred over network during the execution of the protocol. We evaluate this cost as it will lead to some amount of network time of data transmission. We do not evaluate the communication time directly as it is network-dependent.

6.2 Experimental results

6.2.1 Performance on Gowalla

Figure 5 depicts the CPU time and communication cost of three protocols on Gowalla dataset by varying the willing range from 1km to 6km. As seen from the left side of Figure 5, the CPU time of all three protocols increase when the willing range gets large. This is because there are more candidate POIs (and also more users visiting these POIs) in the bigger range, so more scores need to be evaluated over encrypted data. Further, selective publication of check-ins (i.e., KAP or SKAP) is more effective on a larger willing range, as in this case check-ins are more sparse. For example, when the willing range is 6km, the adoption of KAP can save about 67% CPU time. Note that we do not show the performance of PLAS here as it has a considerable amount of CPU time due to the big check-in matrix, but the cloaking area indeed greatly reduces computation cost as shown theoretically before. From the right side of Figure 5, it is clear that all three protocols incur more communication cost as the willing range becomes larger. However, we observe a much slow increase in communication cost when we selectively publish check-ins, as only a small portion of check-ins needs to be published considering the sparsity of check-in data.

In Figure 5, we can also observe that PLAS-CA-SKAP is slightly better than PLAS-CA-KAP. To highlight the difference between them and also to study the impact of parameter k on them, we show in Figure 6 their CPU time and communication cost by varying the value of k. As expected, when k becomes larger, their CPU time and communication cost both increase, as the sub matrices are more sparse and those new zeros introduce more computation and communication. Moreover, PLAS-CA-SKAP is always better than the PLAS-CA-KAP no matter how k changes, and this superiority becomes more significant

[3]https://www.keylength.com/

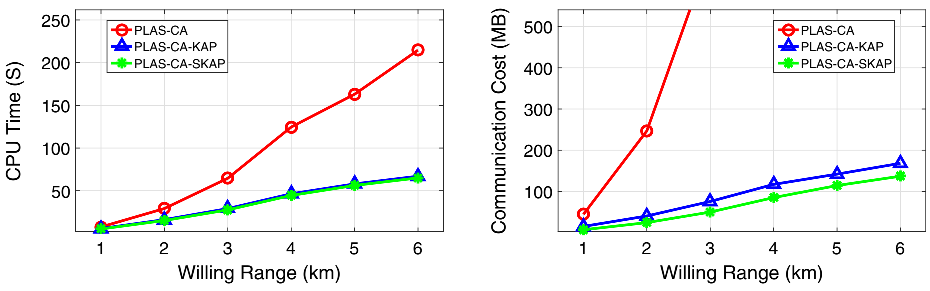


Figure 5 Performance on Gowalla

for large k, which demonstrates that sorting users based on the Hamming distance of their check-ins is useful for publication.

6.2.2 Performance on Yelp

Figure 7 illustrates the CPU time and communication cost of the protocols performing on Yelp dataset by varying the willing range from 1km to 6km. Similar to previous results on Gowalla dataset, the CPU time of all protocols increase as the willing range increases. Also, the k-anonymity publishing protocols have a better performance than PLAS-CA, and all the POI recommendation can be performed within 100 seconds when r = 4km, no matter PLAS-CA, PLAS-CA-KAP and PLAS-CA-SKAP. However, as we can see, due to the extreme increase of users when we increase the willing range, the CPU time of basic protocol also increase a lot, which is especially seen from the range of 5km to 6km. The right figure of Figure 7 shows the communication cost of the proposed protocols on Yelp dataset by varying the willing range from 1km to 6km. Here, we observe the similar communication cost conclusions like in Gowalla dataset, that all protocols have the increasing communication cost with the willing range increasing, especially in PLAS-CA. The k-anonymity publishing protocols can effectively reduce the impact of extreme spareness and have less than 400MB communication cost when the willing range is 6km.

Figure 8 highlights the difference between PLAS-CA-KAP and PLAS-CA-SKAP from CPU time and communication respectively. Adopting the HAMMING DISTANCE makes PLAS-CA-SKAP performing better than PLAS-CA-KAP and the smaller k is, the more unnecessary computation is avoided, which can save computing costs and communication costs.

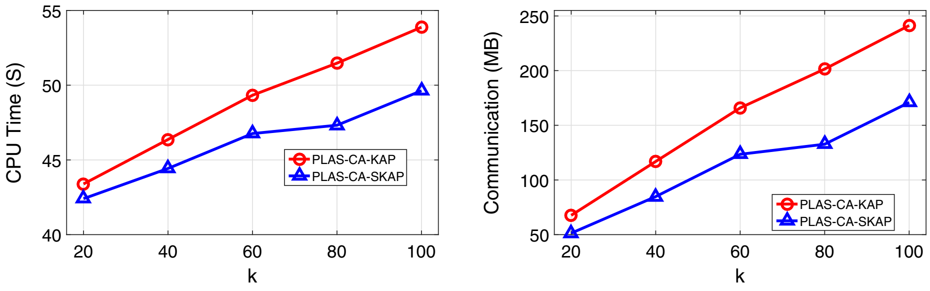


Figure 6 Effect of k on selective publication of Gowalla

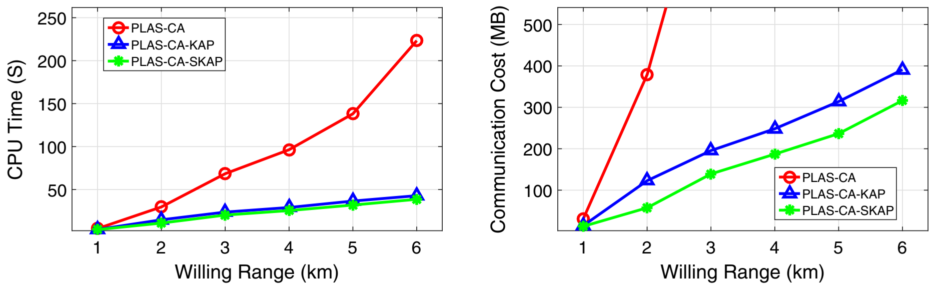


Figure 7 Performance on Yelp

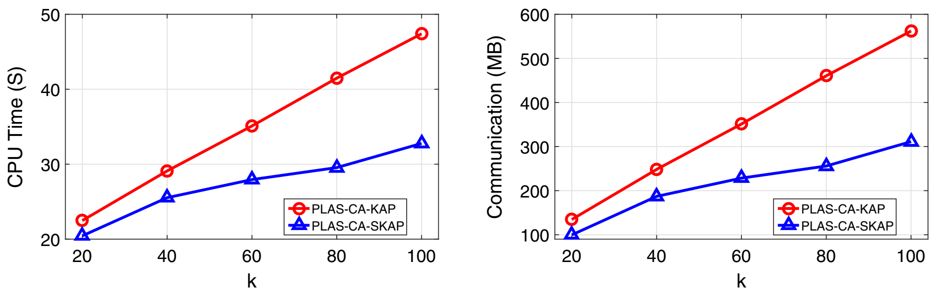


Figure 8 Effect of k on selective publication of Yelp

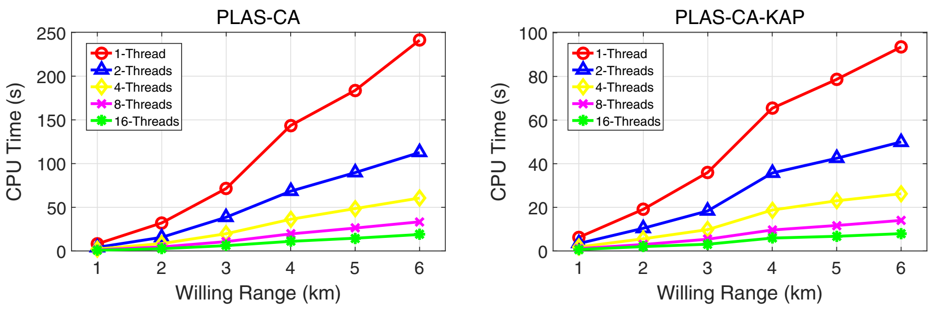


Figure 9 Effect of parallel computing on Gowalla

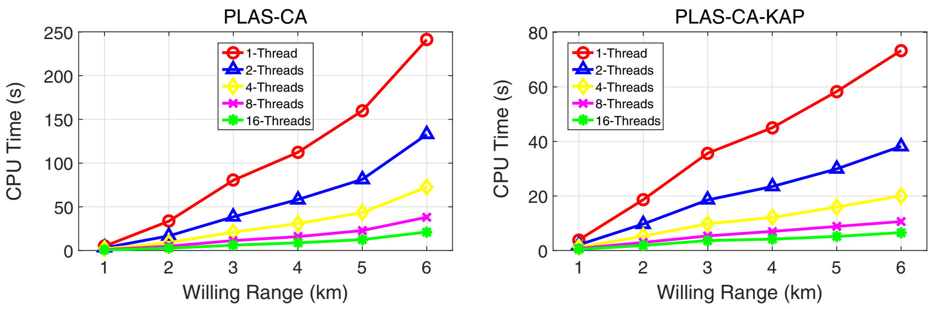


Figure 10 Effect of parallel computing on Yelp

6.2.3 Effects of parallel computing

As mentioned earlier, the expensive operations including encryption, decryption, multiplication, and exponentiation over ciphertext can be parallelized since they are performed on independent data. Therefore the efficiency of the proposed protocols can be improved by parallel computing. Figure 9 shows the performance of PLAS-CA and PLAS-CA-KAP on Gowalla. In the case of one thread, PLAS-CA and PLAS-CA-KAP needs about 240 and 90 seconds, respectively, when the willing range is 6km. When 16 threads are used, their CPU time are roughly 18 and 8 seconds, respectively. For PLAS-CA, the speed up is about 13. For PLAS-CA-KAP, the speed up is about 11. We also test parallel computing on Yelp dataset, and the result is depicted in Figure 10. When single thread is used, PLAS-CA needs about 243 seconds while the PLAS-CA-KAP needs 73 seconds. For the 16 threads case, PLAS-CA needs 20 seconds and PLAS-CA-KAP needs 6.5 seconds. The speed up for them are 12 and 11, respectively. Based on the speed up on two data sets, we can conclude that the proposed protocols have a good scalability.

7 Conclusion

In location-aware social POI recommendation, it is common that the data used for recommendation are distributed among multiple parties and the privacy of every party needs to be respected. To address this challenging problem, we have presented a protocol PLAS which protects private data by semantic secure Paillier and computes the scores of POIs completely over encrypted data using Paillier's homomorphic property. To improve the performance of PLAS, we have also proposed two optimization methods. One is cloaking area based on which geographical-influence can be approximately estimated on plaintext. The other is selective check-ins publication which greatly reduces not only the encryption cost of check-in data but also subsequent computation cost of POI scores over encrypted data. Their effectiveness have been demonstrated by experiments on two real datasets. In summary, our approach is secure, efficient, and can scale to large POI recommendation problems in practice.

Acknowledgements Research reported in this publication was partially supported Natural Science Foundation of China (Grant Nos. 61572336, 61572335, 61402313)

References

1. Bagci, H., Karagoz, P.: Context-aware friend recommendation for location based social networks using random walk: In: Proceedings of the 25th International Conference Companion on World Wide Web, pp. 531–536. ACM (2016)
2. Bamba, B., Liu, L., Pesti, P., Wang, T.: Supporting anonymous location queries in mobile environments with privacygrid. In: Proceedings of the 17th International Conference on World Wide Web, pp. 237–246. ACM (2008)
3. Bettini, C., Wang, X., Jajodia, S.: Protecting privacy against location-based personal identification. In: Proceedings of the 31st Workshop on Secure Data Management, pp. 185–199. Springer (2005)
4. Chen, L., Gao, Y., Xing, Z., Jensen, C.S., Chen, G.: I2RS: A distributed geo-textual image retrieval and recommendation system. Proc. VLDB Endow. **8**(12), 1884–1887 (2015)
5. Chen, L., Gao, Y., Chen, G., Zhang, H.: Metric all-k-nearest-neighbor search. IEEE Trans. Knowl. Data Eng. **28**(1), 98–112 (2016)

6. Cormode, G., Procopiuc, C., Srivastava, D., Shen, E., Yu, T.: Differentially private spatial decompositions. In: Proceedings of the 28th international conference on data engineering, pp. 20–31. IEEE (2012)
7. Erkin, Z., Veugen, T., Toft, T., Lagendijk, R.L.: Generating private recommendations efficiently using homomorphic encryption and data packing. IEEE Trans. Inf. Forens. Secur. **7**(3), 1053–1066 (2012)
8. Gao, H., Tang, J., Liu, H.: Addressing the cold-start problem in location recommendation using geo-social correlations. Data Min. Knowl. Disc. **29**(2), 299–323 (2015)
9. Gao, L., Guo, Z., Zhang, H., Xing, X., Shen, H.T.: Video captioning with attention-based lstm and semantic consistency. IEEE Trans. Multimed. **19**(9), 2045–2055 (2017)
10. Gao, Y., Zheng, B., Chen, G., Lee, W.-C., Lee, K.CK., Li, Q.: Visible reverse k-nearest neighbor query processing in spatial databases. IEEE Trans. Knowl. Data Eng. **21**(9), 1314–1327 (2009)
11. Gedik, B., Liu, L.: Location privacy in mobile systems A personalized anonymization model. In: Proceedings of 25th IEEE International conference on distributed computing systems, pp. 620–629 IEEE (2005)
12. Ghinita, G., Kalnis, P., Khoshgozaran, A., Shahabi, C., Tan, K.-L.: Private queries in location based services: anonymizers are not necessary. In: Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data, pp. 121–132. ACM (2008)
13. Goldreich, O., Ron, D.: On proximity oblivious testing. In: Proceedings of the 41st Annual ACM, Symposium on Theory of Computing, pp. 141–150 (2009)
14. Hosseini, S., Li, L.T.: Point-of-interest recommendation using temporal orientations of users and locations. In: Proceedings of the 21nd international conference on database systems for advanced applications, pp. 330–347. Springer (2016)
15. Huang, J., Qi, J., Yabo, X., Chen, J.: A privacy-enhancing model for location-based personalized recommendations. Distrib. Parallel Datab. **33**(2), 253–276 (2015)
16. Jorgensen, Z., Ting, Y.: A privacy-preserving framework for personalized, social recommendations. In: Proceedings of the 17th International Conference on Extending Database Technology, pp. 571–582 (2014)
17. Li, H., Ge, Y., Hong, R., Zhu, H.: Point-of-interest recommendations Learning potential check-ins from friends. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 975–984. ACM (2016)
18. Liu, B., Hengartner, U.: Privacy-preserving social recommendations in geosocial networks. In: Proceedings of the Eleventh Annual International Conference on Privacy, Security and Trust, pp. 69–76. IEEE (2013)
19. Liu, Y., Wei, W., Sun, A., Miao, C.: Exploiting geographical neighborhood characteristics for location recommendation. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 739–748. ACM (2014)
20. Liu, S., Liu, A., Liu, G., Li, Z., Xu, J., Zhao, P., Zhao, L.: A secure and efficient framework for privacy preserving social recommendation.: In: Proceedings of the 17th Asia-Pacific Web Conference, pp. 781–792. Springer (2015)
21. Liu, A., Zhengy, K., Liz, L., Liu, G., Zhao, L., Zhou, X.: Efficient secure similarity computation on encrypted trajectory data. In: Proceedings of the 31st IEEE international conference on data engineering, pp. 66–77. IEEE (2015)
22. Liu, X., Li, Z., Deng, C., Tao, D.: Distributed adaptive binary quantization for fast nearest neighbor search. IEEE Trans. Image Process. **26**(11), 5324–5336 (2017)
23. Liu, A., Li, Z.-X., Liu, G.-F., Zheng, K., Zhang, M., Li, Q., Zhang, X.: Privacy-preserving task assignment in spatial crowdsourcing. J. Comput. Sci. Technol. **32**(5), 905–918 (2017)
24. Liu, A., Wang, W., Shang, S., Li, Q., Zhang, X.: Efficient task assignment in spatial crowdsourcing with worker and task privacy protection. GeoInformatica, 1–28 (2017)
25. Liu, A., Wang, W., Li, Z., Liu, G., Li, Q., Zhou, X., Zhang, X.: A privacy-preserving framework for trust-oriented point-of-interest recommendation. IEEE Access **6**, 393–404 (2018)
26. McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the netflix prize contenders. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 627–636. ACM (2009)
27. Miao, X., Gao, Y., Chen, G., Cui, H., Guo, C., Pan, W.: SI2P: A restaurant recommendation system using preference queries over incomplete information. Proc. VLDB Endow. **9**(13), 1509–1512 (2016)
28. Mokbel, M.F., Chow, C.-Y., Aref, W.a.G.: The new casper: Query processing for location services without compromising privacy. In: Proceedings of the 32nd International Conference on Very Large Data Bases, pp. 763–774. VLDB Endowment (2006)

29. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of the 18th International Conference on the Theory and Application of Cryptographic Techniques, pp. 223–238. Springer (1999)
30. Rao, Y., Xie, H., Li, J., Jin, F., Lee Wang, F., Li, Q.: Social emotion classification of short text via topic-level maximum entropy model. Inf Manag **53**(8), 978–986 (2016)
31. Rao, Y., Li, Q., Qingyuan, W., Xie, H., Lee Wang, F., Wang, T.: A multi-relational term scheme for first story detection. Neurocomputing **254**, 42–52 (2017)
32. Samanthula, B.K., Rao, F.-Y., Bertino, E., Yi, X.: Privacy-preserving protocols for shortest path discovery over outsourced encrypted graph data. In: Proceedings of the 16th International Conference on Information Reuse and Integration, pp. 427–434. IEEE (2015)
33. Song, J., Shen, H.T., Wang, J., Zi, H., Sebe, N., Wang, J.: A distance-computation-free search scheme for binary code databases. IEEE Trans. Multimed. **18**(3), 484–495 (2016)
34. Song, J., Gao, L., Li, L., Zhu, X., Sebe, N.: Quantization-based hashing: A general framework for scalable image and video retrieval. Pattern Recogn. **75**, 175–187 (2018)
35. Song, J., Zhang, H., Xiangpeng L, Lianli, G., Wang, M., Hong, R.: Self-supervised video hashing with hierarchical binary auto-encoder. IEEE Transactions on Image Processing (2018)
36. Wang, X., Gao, L., Wang, P., Sun, X., Liu, X.: Two-stream 3-d convnet fusion for action recognition in videos with arbitrary size and length. IEEE Trans. Multimed. **20**(3), 634–644 (2018)
37. Wei, L., Hou, J., Yan, Y., Zhang, M., Du, X., Moscibroda, T.: MSQL: Efficient similarity search in metric spaces using SQL. VLDB J **26**(6), 829–854 (2017)
38. Xie, H., Li, X., Wang, T., Li, C.n., Ke, L., Lee Wang, F., Yi, C., Li, Q., Min, H.: Personalized search for social media via dominating verbal context. Neurocomputing **172**, 27–37 (2016)
39. Ye, M., Yin, P., Lee, W.-C., Lee, D.-L.: Exploiting geographical influence for collaborative point-of-interest recommendation. In: Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 325–334. ACM (2011)
40. Ying, H., Chen, L., Xiong, Y., Pgrank, J.W.: Personalized geographical ranking for point-of-interest recommendation. In: Proceedings of the 25th International Conference Companion on World Wide Web, pp. 137–138. ACM (2016)
41. Yuan, Q., Cong, G., Sun, A.: Graph-based point-of-interest recommendation with geographical and temporal influences. In: Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management, pp. 659–668. ACM (2014)
42. Zhang, J.-D., Chow, C.-Y.: Point-of-interest recommendations in location-based social networks. Sigspatial Special **7**(3), 26–33 (2016)
43. Zhu, X., Li, X., Zhang, S., Chunhua, J., Xindong, W.: Robust joint graph sparse coding for unsupervised spectral feature selection. IEEE Trans. Neural Netw. Learn. Syst. **28**(6), 1263–1275 (2017)