SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

September 2017, Vol. 60 092101:1–092101:16
doi: 10.1007/s11432-015-0981-4

APPLET: a privacy-preserving framework for location-aware recommender system

Xindi MA[1,2], Hui LI[2], Jianfeng MA[1,2], Qi JIANG[2], Sheng GAO[3], Ning XI[2*] & Di LU[1]

[1]*School of Computer Science and Technology, Xidian University, Xi'an* 710071*, China;*
[2]*School of Cyber Engineering, Xidian University, Xi'an* 710071*, China;*
[3]*School of Information, Central University of Finance and Economics, Beijing* 102206*, China*

Received March 9, 2016; accepted April 22, 2016; published online October 13, 2016

Abstract Location-aware recommender systems that use location-based ratings to produce recommendations have recently experienced a rapid development and draw significant attention from the research community. However, current work mainly focused on high-quality recommendations while underestimating privacy issues, which can lead to problems of privacy. Such problems are more prominent when service providers, who have limited computational and storage resources, leverage on cloud platforms to fit in with the tremendous number of service requirements and users. In this paper, we propose a novel framework, namely APPLET, for protecting user privacy information, including locations and recommendation results, within a cloud environment. Through this framework, all historical ratings are stored and calculated in ciphertext, allowing us to securely compute the similarities of venues through Paillier encryption, and predict the recommendation results based on Paillier, commutative, and comparable encryption. We also theoretically prove that user information is private and will not be leaked during a recommendation. Finally, empirical results over a real-world dataset demonstrate that our framework can efficiently recommend POIs with a high degree of accuracy in a privacy-preserving manner.

Keywords recommender system, location-based service, homomorphic encryption, privacy-preserving framework, collaborative filtering

Citation Ma X D, Li H, Ma J F, et al. APPLET: a privacy-preserving framework for location-aware recommender system. Sci China Inf Sci, 2017, 60(9): 092101, doi: 10.1007/s11432-015-0981-4

1 Introduction

With the development of urban computing [1] and GPS-enabled devices, location-based services (LBSs) have been widely used, providing us with a convenient way to experience life than ever before. For example, Foursquare allows users to "check-in" at various point-of-interests (POIs) and leave various ratings regarding their experience there. Such ratings have motivated an interesting new paradigm of location-aware recommender systems [2]. Compared with traditional systems, in addition to providing the recommendation ratings (e.g., for Amazon inventory or Netflix titles), location-aware recommender systems have to take into account both spatial-temporal and rating information.

The main challenge for location-aware recommender systems is how to securely and efficiently provide recommendations among a large number of POIs. However, simply applying traditional recommendation

* Corresponding author (email: nxi@xidian.edu.cn)

techniques in an LBS may not be applicable for the following two reasons. First, as spatial data are ubiquitous and highly evolving, traditional recommender systems require a heavy toll for storing and computing user recommendations. Thus, many LBS providers have now turned to untrusted clouds for help. By moving user data and recommendation frameworks to the cloud, service providers can reduce the overhead of their computational resources while preserving the service quality. For example, Netflix shut down the last of its data centers this summer and has moved all of its data to Amazon's cloud [3]. However, this technique has inevitably led to another challenge, namely, privacy. Because user data and recommendation results include a certain amount of privacy information, such as user locations and preferences, the cloud can easily infer who is interested in what and where. Thus, the cloud may track users directly or release their preferential information to advertisers [4]. As a result, users may be afraid that their sensitive information might be leaked to unauthorized attackers, which would be a huge barrier for the development and spread of recommender systems. Therefore, it is crucial to protect user privacy when utilizing a location-aware recommender system. Notably, privacy protection includes not only users' privacy information attached with requests, but also the historical ratings and similarities of venues, which are considered to be the property of the service providers. Unfortunately, until now there have been limited research efforts or valuable contributions regarding this aspect. State-of-the-art work either suffers from an inaccurate recommendation quality [5,6] or low efficiency [7].

In this paper, to address the aforementioned challenges, we propose a privacy-preserving framework for location-aware recommender system (APPLET). For APPLET, we adopt an item-based collaborative filtering algorithm and utilize multiple encryption techniques to help service providers generate recommendations in a privacy-preserving manner. We also show that APPLET is sufficiently secure to protect both user privacy and the profits of the service providers. The main contributions of this paper are as follows.

• We propose a novel framework, namely APPLET, which allows service providers that have moved most of their data to a cloud, to generate recommendations without leaking any user privacy to the cloud itself.

• To reduce the overhead, the service provider would send their historical rating data to the cloud platform. We then utilize the Paillier homomorphic encryption [8] and enable the cloud to compute the venue similarities in ciphertext to protect the service provider's profits. Finally, to protect the users' privacy regarding their location during a recommendation, we employ a comparable encryption [9] such that the cloud platform can filter out those venues that fall into the areas of interest of the users purely through ciphertext. In this way, the computational tasks conducted by the service provider are minimized and sensitive information on both the users and service provider are not leaked to an untrusted cloud.

• We conducted an analysis of APPLET in terms of both theory and practice. The results indicate that APPLET can respond to user requests efficiently and effectively.

The rest of this paper is organized as follows. Section 2 provides the system overview and problem formulation. In Section 3, we present APPLET in detail, followed in Section 4 by an analysis of its security and efficiency. In Section 5, we empirically test the recommendation quality and computational costs. Section 6 presents some related work. Finally, we provide some concluding remarks regarding this paper in Section 7.

2 System overview and problem formulation

As discussed above, simply implementing LBS recommendations over the cloud may invoke unexpected privacy issues, which becomes a key bottleneck for the development and widespread use of recommender systems. To this end, we designed APPLET based on multiple encryption methods to help service providers store and compute their spatial data over a semi-honest cloud environment. In this section, we first present some preliminaries that serve as the basis of APPLET, and then present the system model, threat model, and design goals for APPLET. For the reader's convenience, the notations used in the sequel are listed in Table 1.

Table 1 Definitions and notations used in APPLET

Symbol	Definition
R_t	The rating data collected by SP at time t
R_u	u_q's historical ratings data
pk_s	SP's public key for commutative encryption
sk_s	SP's private key for commutative encryption
PK_s	SP's public key for Paillier encryption
SK_s	SP's private key for Paillier encryption
param	SP's or u_q's parameter for comparable encryption
mkey	SP's or u_q's master key for comparable encryption
pk_u	u_q's public key for commutative encryption
sk_u	u_q's private key for commutative encryption

2.1 Preliminaries

2.1.1 *Collaborative filtering*

Owing to its popularity and widespread adoption in commercial recommender systems (e.g., Amazon), APPLET uses item-based collaborative filtering (CF) as its primary recommendation technique. Through the item-based CF algorithm, the recommender system can produce accurate recommendation results with little computation overhead. Collaborative filtering assumes a set of m users $U = \{u_1, \ldots, u_m\}$, and a set of n venues $V = \{v_1, \ldots, v_n\}(m >> n)$, where each venue has three attributes $A_{vi=}\{v_{iN}, v_{ix}, v_{iy}\}$, in which v_{iN} denotes the ID of v_i, v_{ix} (resp., v_{iy}) denotes the latitude (resp., longitude) of v_i. Each user u_i expresses opinions over a set of venues $V_{ui} \subseteq V$. Opinions can be numeric ratings (e.g., 1 to 5 stars in Netflix), or unary (e.g., "Like/Unlike" for Facebook). Conceptually, ratings are represented as a matrix $R \in \mathbb{R}^{n\times m}$, where each entry $r_{ij} \in R$ denotes the rating posted by user u_j over v_i. Given a requestor u_q, CF produces k recommended venues $v_{s1}, \ldots, v_{sk} \in V$ in which u_q is predicted to like the most.

To achieve this, we should first compute a similarity score $\mathrm{sim}(v_p, v_q)$ for each pair of objects v_p, v_q that have at least one common rating by the same user. The cosine similarity is used to compute the score [2]: $\mathrm{sim}(v_p, v_q) = v_p \cdot v_q/(\|v_p\| \, \|v_q\|)$.

Afterward, using these scores, recommendations are produced by computing the predicted rating of u_q, i.e., $P_{(u_q,i)}$, for each venue i not rated by him [10]: $P_{(u_q,i)} = \sum_{\ell\in\mathcal{L}} \mathrm{sim}(i,\ell) \cdot r_\ell / \sum_{\ell\in\mathcal{L}} |\mathrm{sim}(i,\ell)|$. Notably, each similarity list $\mathcal{L}$ has been reduced to contain only venues rated by u_q.

2.1.2 *Commutative encryption*

Commutative encryption [11] is a useful but rather strict notion of cryptography. In this paper, we adopt commutative encryption to protect the venue attributes (e.g., names and locations) from leaking to service providers when a cloud platform replies to a user's recommendation results. Our commutative encryption is based on the El-Gamal encryption scheme [12] and contains the following six algorithms: KeyGen, Encrypt, Re-encrypt, Decrypt and Re-decrypt.

KeyGen. Generate an efficient description of a cyclic group G of order N with generator g. Choose a random number x_i for each user i from $1, \ldots, N-1$ and compute $y_i = g^{x_i}$. The public key of user i is then denoted as $pk_i = (G, N, g, y_i)$ and the private key is denoted as $sk_i = (G, N, g, x_i)$. Thus for users i and j, their public and private keys are $pk_i = (G, N, g, y_i)$, $pk_j = (G, N, g, y_j)$, $sk_i = (G, N, g, x_i)$, and $sk_j = (G, N, g, x_j)$.

Encrypt. Upon input pk_i and plaintext m, uniformly select an element $k_i \in Z_N$ and output: $(y_1, y_2) = E_{pk_i}(m) = (g^{k_i} (\mathrm{mod}\ N), m y_i^{k_i} (\mathrm{mod}\ N))$.

Re-encrypt. With a ciphertext (y_1, y_2) and pk_j, uniformly select an element $k_j \in Z_N$ and output: $(c_1, c_2, c_3) = E_{pk_j}(y_1, y_2) = (y_1, g^{k_j} (\mathrm{mod}\ N), y_2 y_j^{k_j} (\mathrm{mod}\ N))$.

Decrypt. Upon input sk_i and ciphertext (c_1, c_2, c_3), output: $(y_1', y_2') = D_{sk_i}(c_1, c_2, c_3) = (c_2, c_3 (c_1^{x_i})^{-1} (\mathrm{mod}\ N))$.

Re-decrypt. Upon input sk_j and ciphertext (y_1', y_2'), output: $D_{sk_j}(y_1', y_2') = y_2' (y_1'^{x_j})^{-1} (\mathrm{mod}\ N) = m$.

2.1.3 *Paillier homomorphic encryption*

To ensure that the ratings can be calculated in the form of a ciphertext in a cloud platform, we adopt Paillier encryption to encrypt the historical ratings. In this way, the cloud platform cannot obtain the properties of the service providers. After being encrypted by the Paillier encryption scheme, the ciphertexts satisfy the following properties.

• The product of two ciphertexts will decrypt to the sum of their corresponding plaintexts: $D_{SK_i}(E_{PK_i}(x) \cdot E_{PK_i}(y)) = D_{SK_i}(E_{PK_i}(x+y)) = x+y$.

• Given constant number $a \in \mathbb{Z}_N$ and ciphertext $E_{PK_i}(x)$, we have the following: $D_{SK_i}(E_{PK_i}(x)^a) = D_{SK_i}(E_{PK_i}(a \cdot x)) = a \cdot x$.

2.1.4 *Comparable encryption*

Comparable encryption [9] aims to overcome the weakness of order-preserving encryption (OPE), as the comparison between data samples is frequently executed in the range queries. Through a comparable encryption, the results can be attained by executing a query within a single interaction while achieving the security of the weak indistinguishability defined in [9]. In this paper, we adopt comparable encryption to compare the locations of venues with users' requesting areas in the ciphertext. Through comparable encryption, the cloud platform can filter out the venues which located in users' interesting area while not obtaining users' locations. Given a plaintext num, comparable encryption encrypts it by Der and Enc functions and produces a token, token, and ciphertext, ciph. Given two ciphertexts {ciph, ciph$'$} encrypted through comparable encryption, we can compare their numerical order as follows:

$$\text{token} = \text{Der}(\text{param}, \text{mkey}, \text{num}), \quad \text{ciph} = \text{Enc}(\text{param}, \text{mkey}, \text{num}), \quad \text{ciph}' = \text{Enc}(\text{param}, \text{mkey}, \text{num}'),$$

$$\text{Cmp}\,(\text{param}, \text{ciph}, \text{ciph}', \text{token}) = \begin{cases} 0 \text{ if num} = \text{num}', \\ 1 \text{ if num} > \text{num}', \\ 2 \text{ if num} < \text{num}'. \end{cases}$$

2.2 System model

Before describing APPLET, we formally present the definition of a general location-aware recommender system as follows.

Definition 1. Given the location (x_u, y_u) of user u_q as well as the requested distance threshold $(\Delta x, \Delta y)$ of the POIs, a location-aware recommender system returns the predicted ratings of u_q, i.e., R_p, of venues located in u_q's interesting area defined by $(x_u \pm \Delta x, y_u \pm \Delta y)$, by taking into account the ratings for visited venues of u_q, i.e., $R_u \in \mathbb{R}^n$.

To guarantee the privacy of such a recommendation in a cloud environment, all privacy information (i.e., (x_u, y_u), $(\Delta x, \Delta y)$, and R_p) should be kept and computed in ciphertext. In this manner, we can derive the basic components for our privacy-preserving location-aware recommender system as follows (see Figure 1):

• Trusted authority (TA). TA is an indispensable entity that is trusted by all entities. It distributes and manages all private keys involved in the framework.

• Cloud platform (CP). CP stores and manages all data in the framework. In addition, CP has to conduct many calculations over the stored data, especially the recommendation algorithm.

• Service provider (SP). SP owns the venue attributes and collects the correlated historical ratings from each user at regular intervals. However, SP possesses limited storage and computational resources, and thus it sends the collected ratings to CP regularly for storage and computation. Finally, with the help of CP, SP can compute the similarities among venues based on all of the ratings data stored in CP.

• Recommendation users (RUs). RUs send their locations and distance thresholds to CP for recommendations when they request a recommendation service.

Given these parties, APPLET works as follows:

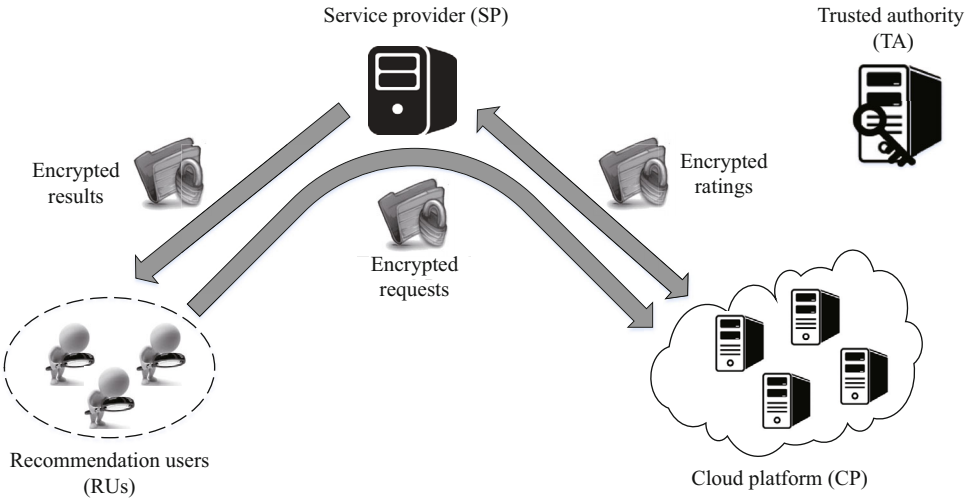

Figure 1 System Framework of APPLET.

• SP sends historical ratings R_t and attributes A_v to CP regularly. SP then computes the similarities of the venues $\text{Sim} \in \mathbb{R}^{n \times n}$ with the help of CP.

• Whenever requesting a recommendation, an RU u_q sends his location $(x_u,\ y_u)$ and the distance threshold of POIs $(\Delta x, \Delta y)$ to CP.

• CP filters out the venues $V' \in V$ in the area bounded by the threshold, and sends the aggregated results to SP.

• Based on the results from CP, SP computes the predicted ratings R_p and sends $\{(v'_N, v'_L), R_p\}$ to u_q.

Notably, most of the calculations are executed on CP. SP and RUs are only responsible for a very limited number of decryption tasks to obtain the results. During the entire procedure, sensitive information should be kept secure against internal and external parties (e.g., CP and adversaries). We then explicitly list the threat model and all design goals that APPLET should satisfy.

2.3 Threat model

In APPLET, SP is curious-but-honest that is interested in u_q's recommendation results R_p at this time but provides correct historical rating data R_t that are collected at time t. The third party CP is also considered as curious-but-honest in two aspects. First, CP is interested in SP's historical rating data, R, $(R = R_1 \bigcup R_2 \bigcup \cdots \bigcup R_t)$, and similarities, Sim. Second, CP is also curious about u_q's recommendation results, R_p, and locations. Notably, CP strictly follows the protocols executed in the framework. Moreover, an external adversary is interested in all data transmitted in the framework by eavesdropping.

2.4 Design goals

As a privacy-preserving location-aware recommender system, APPLET should fulfill the following requirements.

• **Recommendation quality.** The proposed framework should return the effective and satisfactory results of any location-aware recommendation requests from RUs.

• **Recommendation efficiency.** The proposed framework should return the recommendation results efficiently to provide a better user experience.

• **Security goals.** The proposed framework should achieve the privacy requirements in terms of the following aspects:

(1) Privacy of SP. When the similarity computation ends, CP obtains the processed similarities of venues F and the encrypted attributes. The adversary and CP should learn nothing beyond F and these encrypted attributes.

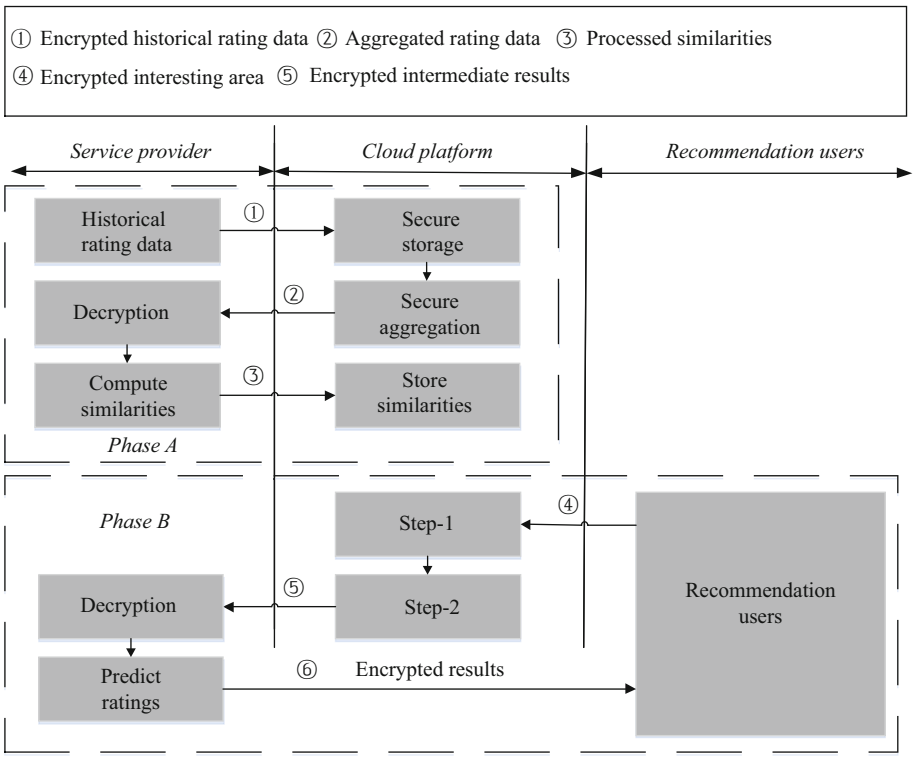

Figure 2 Overall APPLET procedure.

(2) Privacy of RUs. When the prediction ends, RUs obtain the recommendation results R_p. The adversary, CP, and SP should learn nothing beyond what can be derived from their outputs and private inputs in the protocol.

3 Construction of APPLET framework

In this section, we present the details of APPLET, which mainly consists of two phases: privacy-preserving similarities computation (Phase A) and privacy-preserving ratings prediction (Phase B). The overall procedure is shown in Figure 2.

3.1 Privacy-preserving similarities computation (Phase A)

In this phase, SP computes the similarities of the venues and uploads them along with the attributes to CP. To protect such private information, the ratings R_t and attributes A_v are encrypted and sent to CP in a ciphertext. Notably, we have to guarantee the recoverability of A_v under the premise of comparing their locations with user's interesting area. Thus, we extend the tuple $A_v = \{v_N, v_x, v_y\}$ into $A'_v = \{(v_N, v_x, v_y), (v_x, v_y)\} = \{(v_N, v_L), v_L\}$. The attributes A_v and extended locations $\{v_L\}$ are then encrypted through commutative encryption [13] and comparable encryption [9,14], respectively. Moreover, we construct an efficient and privacy-preserving protocol in Pailliar homomorphic encryption [8] to compute the venue similarities, Sim, securely.

3.1.1 *Encrypting the venue attributes*

For all venues, their extended attributes can be denoted as $\{(v_N, v_L), v_L\}$. We adopt commutative encryption and comparable encryption to allow SP to encrypt $\{v_N, v_L\}$ and $\{v_L\}$, respectively. SP then sends ciphertext $\{E_{pk_s}(v_N, v_L), \mathrm{Enc}(v_L)\}$ to CP.

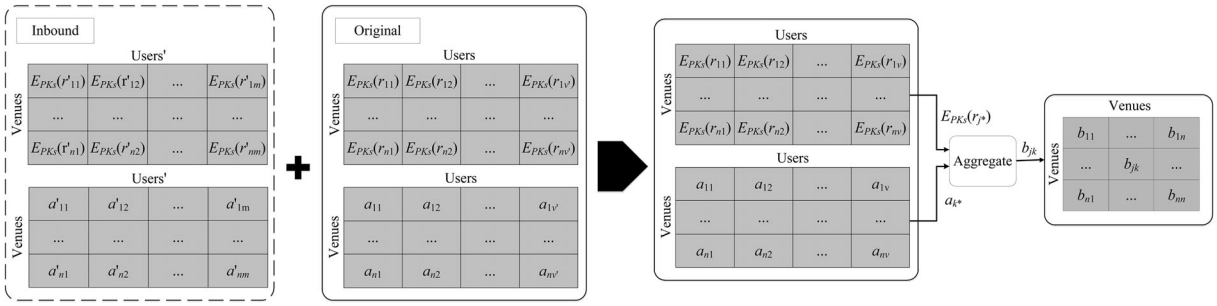

Figure 3 The aggregation of venues (@CP).

3.1.2 *Similarity computing protocol*

Because the historical rating data are considered as assets of SP and contain the private information of users, such data must be calculated in the form of a ciphertext by CP. The similarity computing protocol is thus as follows.

Step-I(@SP). We encrypt R_t in Paillier encryption using the public key of SP, PK_s, producing $E_{PK_s}(R_t)$. In addition, we also process a copy of R in the following way [15].

Given security parameters k_1, k_2, a large prime p, and another large prime α, such that $|\alpha| = k_1$, a large random number $s \in Z_p$, and $n \cdot m$ random numbers c_{ij}, $i \in [1, n], j \in [1, m]$, with $|c_{ij}| = k_2$. For each $r_{ij} \in R_t$, calculate $a_{ij} = s \cdot (r_{ij} \cdot \alpha^2 + c_{ij}) \bmod p$, resulting in a new matrix, $A_t \in \mathbb{R}^{n \times m}$. Afterward, SP keeps $s^{-1} \bmod p$ and α secret, and sends $\{E_{PK_s}(R_t), A_t\}$ to CP.

Step-II(@CP). When receiving $\{E_{PK_s}(R_t), A_t\}$, CP integrates the inbound data $\{E_{PK_s}(R_t), A_t\}$ into the original historical rating data $\{E_{PK_s}(R), A\}, (R \in \mathbb{R}^{n \times v'})$, which has been stored here, and calculates the aggregated ciphertext as follows, $i \in [1, v], j, k \in [1, n], (v \gg m \gg n, v = m + v')$: $b_{jk} = \Pi_{i=1}^{v} E_{PK_s}(r_{ji})^{a_{ki}} \bmod N^2 = E_{PK_s}(\sum_{i=1}^{v} r_{ji} \cdot s(r_{ki} \cdot \alpha^2 + c_{ki}) \bmod p) \bmod N^2$.

The detailed process of this is shown in Figure 3, and the matrix of the aggregated results is represented as $B \in \mathbb{R}^{n \times n}$. In this way, the dimensions of the rating data are reduced from $n \times v$ to $n \times n$. CP then sends B to SP for the remaining computation. As $n \ll m \ll v$, the space requirements for SP are significantly reduced.

Step-III(@SP). After receiving aggregated matrix B, SP decrypts it and produces the decrypted aggregated result b'_{jk}: $b'_{jk} = D_{SK_s}(b_{jk}) = \sum_{i=1}^{v} r_{ji} \cdot s \cdot (r_{ki} \cdot \alpha^2 + c_{ki}) \bmod p$.

Then, SP will conduct the following for $j, k \in [1, n]$: $d_{jk} = \frac{s^{-1} \cdot b'_{jk} - s^{-1} \cdot b'_{jk} \bmod \alpha^2}{\alpha^2} = \sum_{i=1}^{v} r_{ji} r_{ki}$. Thus, the venue similarities are $\mathrm{sim}_{jk} = d_{jk} / (\sqrt{d_{jj}} \cdot \sqrt{d_{kk}})$.

Suffering from limited computation resources, SP sends the similarities to CP to respond to the requests of RUs quickly. However, the similarities are considered the private property of SP. Hence, SP needs to preprocess the similarities before outsourcing. In APPLET, the preprocess is as follows:

Choose $n \cdot n$ random numbers w_{jk} for $j, k \in [1, n]$, with $|w_{jk}| = k_2$. For each sim_{jk}, calculate $f_{jk} = s \cdot (\mathrm{sim}_{jk} \cdot \alpha^2 + w_{jk}) \bmod p$.

All f_{jk} form a new matrix $F \in \mathbb{R}^{n \times n}$. Afterward, SP sends F to CP. All processes of Phase A are listed in Algorithm 1. In order to obtain the correct result, we define the following constraints:

$$\sum_{i=1}^{v} r_{ji} \cdot (r_{ki} \cdot \alpha^2 + c_{ki}) < p \text{ and } \sum_{i=1}^{v} r_{ji} \cdot c_{ki} < \alpha^2.$$

3.2 Privacy-preserving ratings prediction (Phase B)

Whenever a request is issued, the area of interest and three secure parameters are uploaded to CP for recommendation. However, such information may disclose the user's privacy. Thus, RUs must encrypt their area of interest in advance.

Step-I(@RUs). Before sending a request to CP, the requestor u_q will generate the tuples as follows:

Algorithm 1 Privacy-preserving similarity computation

Input: SP has an $n \cdot m$ historical rating data matrix R_t, public key PK_s, private key SK_s, security parameters k_1, k_2, and a large prime p, and CP holds the original encrypted rating data and the processed copy data.

Output: CP obtains the processed venue similarities.

(@SP):
Make a copy of R_t.
Choose a large prime α such that $|\alpha| = k_1$, and choose a large random number, $s \in Z_p$.
for $i \in [1, n]$ and $j \in [1, m]$ **do**
  Encrypt r_{ij} through PK_s to obtain $E_{PK_s}(r_{ij})$.
  Choose a random number c_{ij} with $|c_{ij}| = k_2$ and for each copy r_{ij}, calculate $a_{ij} = s \cdot (r_{ij} \cdot \alpha^2 + c_{ij}) \bmod p$.
end for
Send $\{E_{PK_s}(R_t), A_t\}$ to CP.
(@CP):
Integrate the inbound data $\{E_{PK_s}(R_t), A_t\}$ into the original historical data $\{E_{PK_s}(R), A\}$ stored in CP.
Aggregate the historical rating data as follows: $B = \Pi_{i=1}^{v} E_{PK_s}(R)^A \bmod N^2$.
Send the aggregated result matrix B to SP.
(@SP):
Decrypt matrix B to obtain $B' = D_{SK_s}(B)$.
for $j, k \in [1, n]$ **do**
  Calculate $d_{jk} = \frac{s^{-1} \cdot b'_{jk} - s^{-1} \cdot b'_{jk} \bmod \alpha^2}{\alpha^2}$.
end for
for $j, k \in [1, n]$ **do**
  Calculate $\mathrm{sim}_{jk} = d_{jk} / (\sqrt{d_{jj}} \cdot \sqrt{d_{kk}})$.
  Choose a random number w_{jk} with $|w_{jk}| = k_2$ and calculate $f_{jk} = s \cdot (\mathrm{sim}_{jk} \cdot \alpha^2 + w_{jk}) \bmod p$.
end for
return matrix F to CP.

• For the area of interest, the comparable encryption is used to encrypt $\{x_u \pm \Delta x, y_u \pm \Delta y\}$, producing $\mathrm{Enc}(x_u \pm \Delta x, y_u \pm \Delta y)$ and $\mathrm{Der}(x_u \pm \Delta x, y_u \pm \Delta y)$.

• When registering as a recommendation user from SP, u_q will receive the parameter α sent by SP. Then, given the security parameters k_1, k_2, and a large prime p, u_q chooses two other large primes, β and γ, such that $|\beta| = |\gamma| = k_1$ and a large random number $s' \in Z_p$.

Finally, u_q sends $\{\mathrm{Enc}(x_u \pm \Delta x, y_u \pm \Delta y), \mathrm{Der}(x_u \pm \Delta x, y_u \pm \Delta y), s', \beta^2, \alpha^2\gamma\}$ to CP through a secure channel.

Step-II(@CP). Once $\{\mathrm{Enc}(x_u \pm \Delta x, y_u \pm \Delta y), \mathrm{Der}(x_u \pm \Delta x, y_u \pm \Delta y), s', \beta^2, \alpha^2\gamma\}$ are received, CP filters the venues that are located in the area of interest of u_q. CP will then aggregate these venues with the rating data of u_q and send the intermediate results to SP.

• **Filtering out the venues.** First, CP traverses all of the venues to filter out those located in the area of interest of u_q. For each venue v_i, if the following conditions are satisfied, CP will append it to H:

$$\begin{cases} \mathrm{Cmp}(\mathrm{param}, \mathrm{Enc}(x_u - \Delta x), \mathrm{Enc}(v_{ix}), \mathrm{Der}(x_u - \Delta x)) = 2 \text{ or } 0, \\ \mathrm{Cmp}(\mathrm{param}, \mathrm{Enc}(x_u + \Delta x), \mathrm{Enc}(v_{ix}), \mathrm{Der}(x_u + \Delta x)) = 1 \text{ or } 0, \\ \mathrm{Cmp}(\mathrm{param}, \mathrm{Enc}(y_u - \Delta y), \mathrm{Enc}(v_{iy}), \mathrm{Der}(y_u - \Delta y)) = 2 \text{ or } 0, \\ \mathrm{Cmp}(\mathrm{param}, \mathrm{Enc}(y_u + \Delta y), \mathrm{Enc}(v_{iy}), \mathrm{Der}(y_u + \Delta y)) = 1 \text{ or } 0. \end{cases}$$

In this way, all venues in set H are those located in the area of interest of u_q, and we denote $|H| = h$. CP then re-encrypts the encrypted attributes of these venues using the public key of u_q, pk_u, to obtain $E_{pk_u}(E_{pk_s}(v'_N, v'_L))$.

• **Aggregating venues.** After filtering out the venues, CP needs to predict the ratings of these venues by u_q. Since CP does not know the plaintext of the similarities or the historical ratings of u_q, only the intermediate results of the prediction can be calculated in CP, which is as follows:

- Select the venues that u_q has rated and extract his encrypted ratings as $E_{PK_s}(R_u)$, $|R_u| = \mathcal{L}$. Each element in R_u, shown as r_ℓ, represents the opinion of u_q regarding venue ℓ.

- Then, choose $h \cdot \mathcal{L}$ random numbers $z_{i\ell}, i \in [1,h], \ell \in [1,\mathcal{L}]$, and for each venue in H, calculate

$$t_i = \sum_{\ell=1}^{\mathcal{L}} f_{i\ell} = \sum_{\ell=1}^{\mathcal{L}} s(\mathrm{sim}_{i\ell} \cdot \alpha^2 + w_{i\ell}) \bmod p,$$

$$q_{ui} = \prod_{\ell=1}^{\mathcal{L}} E_{PK_s}(r_\ell)^{s'(f_{i\ell}\beta^2 + z_{i\ell}\alpha^2\gamma)} \bmod N^2$$

$$= E_{PK_s}\left(\sum_{\ell=1}^{\mathcal{L}} s'(s(\mathrm{sim}_{i\ell} r_\ell \alpha^2\beta^2 + w_{i\ell} r_\ell \beta^2) + r_\ell z_{i\ell}\alpha^2\gamma)\right) \bmod N^2.$$

The intermediate results are denoted as $T \in \mathbb{R}^h$ and $Q \in \mathbb{R}^h$. CP then sends $\{E_{pk_u}(E_{pk_s}(v'_N, v'_L)), T, Q\}$ to SP.

Step-III(@SP). When receiving the information $\{E_{pk_u}(E_{pk_s}\ (v'_N, v'_L)), T, Q\}$, SP operates as follows:

• For the encrypted attributes of the venues in H, SP decrypts the inner encryption of $\{E_{pk_u}(E_{pk_s}(v'_N, v'_L))\}$ using the private key sk_s of the commutative encryption: $D_{sk_s}(E_{pk_u}(E_{pk_s}(v'_N, v'_L))) = E_{pk_u}(v'_N, v'_L)$.

• For T and Q, SP conducts the following for $i \in [1,h]$: $r'_{pi} = \frac{s^{-1}\cdot D_{SK_s}(q_{ui}) - s^{-1}\cdot D_{SK_s}(q_{ui}) \bmod \alpha^2}{s^{-1}\cdot t_i - s^{-1}\cdot t_i \bmod \alpha^2} = \sum_{\ell=1}^{\mathcal{L}} s'(\mathrm{sim}_{i\ell} r_\ell \beta^2 + s^{-1} r_\ell z_{il}\gamma) \bmod p / \sum_{\ell=1}^{\mathcal{L}} \mathrm{sim}_{i\ell}$.

Because SP cannot absolutely decrypt the attributes of the venues in H, the location and preference privacy of u_q are not leaked to SP. Moreover, SP does not know $s'^{-1} \bmod p$ or β, and thus cannot calculate the predicted ratings either. The encrypted recommender results are represented as R'_p, and SP sends $\{E_{pk_u}(v'_N, v'_L), R'_p\}$ to u_q for the remaining process.

Step-IV(@RUs). After receiving $\{E_{pk_u}(v'_N, v'_L), R'_p\}$, u_q will conduct the following for $i \in [1,h]$:

• For the encrypted attributes $E_{pk_u}(v'_N, v'_L)$, u_q decrypts them using the private key sk_u of the commutative encryption: $D_{sk_u}(E_{pk_u}(v'_N, v'_L)) = \{v'_N, v'_L\}$.

• The predicted ratings for the venues in H can be computed as follows: $r_{pi} = \frac{s'^{-1} r'_{pi} - s'^{-1} r'_{pi} \bmod \beta^2}{\beta^2} = \sum_{\ell=1}^{\mathcal{L}} \mathrm{sim}_{i\ell} r_\ell / \sum_{\ell=1}^{\mathcal{L}} \mathrm{sim}_{i\ell}$.

Afterward, u_q will select the top-k venues ranked by r_{pi}. The detailed process in Phase B is listed in Algorithm 2. To guarantee the result, we define the following constraints:

$$\begin{cases} \sum_{\ell=1}^{\mathcal{L}} s'(\mathrm{sim}_{i\ell} r_\ell \alpha^2 \beta^2 + w_{i\ell} r_\ell \beta^2 + s^{-1} r_\ell z_{i\ell} \alpha^2 \gamma) < p, \\ \sum_{\ell=1}^{\mathcal{L}} s' \cdot w_{i\ell} \cdot r_\ell \cdot \beta^2 < \alpha^2, \\ \sum_{\ell=1}^{\mathcal{L}} s^{-1} \cdot r_\ell \cdot z_{i\ell} \cdot \gamma / \sum_{\ell=1}^{\mathcal{L}} \mathrm{sim}_{i\ell} < \beta^2. \end{cases}$$

4 Security and efficiency analysis

In this section, we theoretically show that APPLET fulfills the security and efficiency requirements illustrated in Subsection 2.4.

4.1 Security analysis

To study the security of APPLET, we adopt a simulation model [16, 17] that is defined in secure two-party protocols for semi-honest adversaries, and widely used to prove the security of multi-party protocols. Intuitively, we say a protocol is secure if each party participating in it can be computed based on its input and output only. We require that a party's view in a protocol execution be simulated only when the input and output are given. This implies that the parties learn nothing from the execution of the protocol itself.

Theorem 1. Both Phase A and Phase B in APPLET are secure in curious-but-honest model.

Proof. The proof is given in Appendix A.

Algorithm 2 Privacy-preserving ratings prediction

Input: User u_q in RUs has area of interest $\{x_u \pm \Delta x, y_u \pm \Delta y\}$, private key sk_u, comparable encryption parameters param, master key mkey, SP's secure parameter α, security parameters k_1 and a large prime p. CP has processed similarities F, encrypted ratings $E_{PK_s}(R)$, u_q's public key pk_u, encrypted attributes of venues $\{E_{pk_s}(v_N, v_L), \mathrm{Enc}(v_L)\}$, comparable encryption parameters param and security parameters k_2. SP holds private key SK_s, sk_s, $s^{-1} \bmod p$, and α.

Output: u_q obtains the recommendation results.

(@RUs):
Generate $\mathrm{Enc}(x_u \pm \Delta x, y_u \pm \Delta y)$ and $\mathrm{Der}(x_u \pm \Delta x, y_u \pm \Delta y)$ using param, mkey.
Choose two large primes β and γ such that $|\beta| = |\gamma| = k_1$, and choose a large random number $s' \in Z_p$.
Send $\{\mathrm{Enc}(x_u \pm \Delta x, y_u \pm \Delta y), \mathrm{Der}(x_u \pm \Delta x, y_u \pm \Delta y), s', \beta^2, \alpha^2\gamma\}$ to CP.
(@CP):
Step-1 (Filtering out venues)
Select the venues located in $(x_u \pm \Delta x, y_u \pm \Delta y)$ through comparable encryption and append them to H.
Encrypt $E_{pk_s}(v'_N, v'_L)$ in H by pk_u, obtaining: $E_{pk_u}(E_{pk_s}(v'_N, v'_L))$.
Step-2 (Aggregating venues)
Select the venues rated by u_q and extract his encrypted ratings: $E_{PK_s}(R_u)$.
for $i \in [1, h]$ **do**
    Choose $h \cdot \ell$ random numbers $z_{i\ell}$ with $|z_{i\ell}| = k_2$ and aggregate the similarities with $E_{PK_s}(R_u)$ as follows: $t_i = \sum_{\ell=1}^{\mathcal{L}} f_{i\ell}$, $q_{ui} = \prod_{\ell=1}^{\mathcal{L}} E_{PK_s}(r_\ell)^{s'(f_{i\ell}\beta^2 + z_{i\ell}\alpha^2\gamma)} \bmod N^2$.
end for
Send $\{E_{pk_u}(E_{pk_s}(v'_N, v'_L)), T, Q\}$ to SP.
(@SP):
Decrypt the inner encryption of $E_{pk_u}(E_{pk_s}(v'_N, v'_L))$ using sk_s: $D_{sk_s}(E_{pk_u}(E_{pk_s}(v'_N, v'_L))) = E_{pk_u}(v'_N, v'_L)$.
Then, $R'_p = \frac{s^{-1} \cdot D_{SK_s}(Q) - s^{-1} \cdot D_{SK_s}(Q) \bmod \alpha^2}{s^{-1} \cdot T - s^{-1} \cdot T \bmod \alpha^2}$.
Send $\{E_{pk_u}(v'_N, v'_L), R'_p\}$ to u_q.
(@RUs):
Decrypt the encrypted attributes $E_{pk_u}(v'_N, v'_L)$ by sk_u for $i \in [1, h] : D_{sk_u}(E_{pk_u}(v'_N, v'_L)) = \{v'_N, v'_L\}$.
Calculate $R_p = (s'^{-1} \cdot R'_p - s'^{-1} \cdot R'_p \bmod \beta^2)/\beta^2$.
return top-k venues ranked by R_p and draw them on the map.

4.2 Efficiency analysis

In this section, we study the communication costs and storage overhead in APPLET. The security parameter of the encryption techniques used is 1024 bits in size. At the beginning of Phase A, all historical ratings and attributes of the venues should be sent to CP, which costs $O(m \cdot n)$ to transmit. CP then aggregates the venues and spends $O(n \cdot n)$ to transmit the aggregated matrix to SP. Afterward, SP computes the similarities of n venues and sends them to CP, which costs $O(n \cdot n)$. Thus, the total communication cost of Phase A is $O(n \cdot (m + 2n))$. In addition, each ciphertext tuple in Phase A requires 1024-bit to be stored. Hence, in Phase A, it costs SP $2n \cdot (m + n + 1) \cdot 1024$-bit to store all of the data. In all, another $2n \cdot (m + n + 1) \cdot 1023$-bit is needed in SP owing to the adoption of the privacy-preserving technique. Notably, a traditional scheme without any privacy-preserving technique also requires $O(n \cdot (m + 2n))$ communication overhead in Phase A.

When u_q requests a recommendation service in Phase B, only his area of interest and three parameters will be sent to CP, which costs $O(1)$ to transmit. After filtering, CP spends $O(h)$ to transmit the selected venues to SP. Additionally, SP also spends $O(h)$ to send the encrypted recommendation results to u_q. Thus, the total communication cost of Phase B is $O(2h)$. In addition, CP also spends $3h \cdot 1024$-bit to store the aggregated results, and SP spends $2h \cdot 1024$-bit to store the recommendation results. Thus, it costs $5h \cdot 1024$-bit to store the ciphertext in Phase B. Hence, such storage costs an extra $5h \cdot 1023$-bit owing to the use of the encryption technique. Notably, the total communication cost of a traditional scheme without any privacy-preserving technique for Phase B is $O(2h)$.

5 Performance evaluations

In this section, we present a series of empirical results of APPLET conducted over a real-world dataset, which indicate that APPLET can effectively and efficiently fulfill the design goals described in Subsection 2.4. APPLET was implemented using Java and MySQL. The experiments were conducted on a

machine with a 2.66 GHz quad-core processor and 8 GB of RAM, along with an Android simulator with an Intel Atom (x86) CPU and 2 GB of RAM.

Dataset. We adopted a real data set consisting of user ratings expressed as one to five stars for spatial venues derived from Foursquare user histories. The Archive Team extracted the data from the Foursquare application through the public API, which owns 2809581 location-based ratings from 2153471 users for 1143092 venues in the State of Minnesota from July 2012 to September 2013 [18].

5.1 Recommendation quality

To measure the quality, we first randomly selected some users and extracted their ratings for 50% of the venues, which are viewed as the ground-truth. We then computed the similarities of the venues using the remaining dataset. Simultaneously, we selected an area as the user's area of interest. Finally, we simulated the recommendation process using this area of interest as input.

To evaluate the prediction accuracy, we focus on how many locations which have the same predicted ratings with the ground-truth appear in the recommendation results. So we use the probabilistic method [19] to measure the recommendation quality, which was evaluated based on the probability of difference between the predicted ratings and the ground-truth: $P_d = \sum_{i=1}^{h} (|r_{pi} - r_i| = d)/h$, where $d = 0, \ldots, 5$. In the equation above, h represents the number of venues located in the user's area of interest; r_{pi} represents the predicted rating for venue v_i; r_i represents the veritable rating for venue v_i; and d represents the difference between the predicted rating and the ground-truth. The simulation results are shown in Figure 4(a). In Figure 4(a), the x-axis shows the recommendation error d, and the y-axis shows the numbers and probabilities (%) of d. Clearly, the probability of $d = 0$ is as high as 78.3%. At $d = 4$ and $d = 5$, P_d is only 7.5% and 3.8%, respectively. This indicates that the recommendation quality of APPLET is sufficiently high to meet the user requirements.

5.2 Recommendation efficiency

To test all factors affecting the efficiency of our APPLET, we randomly selected 1051 location-based ratings from 90 users for 40 venues. In APPLET, there are four factors that affect the efficiency: the number of users in $R : v$, the number of venues in $R : n$, the number of ratings in $R_u : \mathcal{L}$, and the distance threshold (1°) of the POIs: $(\Delta x, \Delta y)$.

During the simulation, we evaluated the run time (including the total time for Phase A and the recommender time for Phase B, the preprocessing time for Phases A and B, and the decryption time of u_q for Phase B) of APPLET using varying factors. In Figures 4(b) and (c), we plot the run time when varying v. Only the total time of Phase A clearly increased, and Phase B was not affected. This is because Phase A needs to calculate more samples for the similarities, which requires more computational resources, whereas Phase B is not affected. As a preprocess, it is believed that the run time of Phase A is reasonable, especially when deploying it on a cloud.

In Figures 4(d) and (e), we plot the run time by varying n. The simulation results show that the total time for Phase A and the recommender time for Phase B clearly increase with n. As n increases, SP and CP need more computational resources to compute the similarities in Phase A. Additionally, in Phase B, CP traverses more venues to filter out those venues that are located in the area of interest of u_q. Then, with an increase in h, CP and SP spend more time to compute the intermediate results, and u_q will also need to spend more time decrypting the ciphertext. Nevertheless, the results show that the efficiency is also sufficiently high to meet the demands of u_q.

In Figures 4(f) and (g), we plot the run time by varying $\mathcal{L}$. The results show that the run time of Phase A substantially remains unchanged, because R is not affected by $\mathcal{L}$. On the other hand, when varying $\mathcal{L}$, CP needs to aggregate more ratings in Phase B. However, the entire recommender time in Phase B only has a slight change, and because h is unchanged, the decryption time of u_q also remains constant.

In Figures 4(h) and (i), we plot the run time by varying $(\Delta x, \Delta y)$. Figure 4(h), the total and preprocess time of Phase A remain constant, because R is not affected by $(\Delta x, \Delta y)$. From Figure 4(h), the recommender time of Phase B and the decryption time of u_q clearly increase with $(\Delta x, \Delta y)$. The

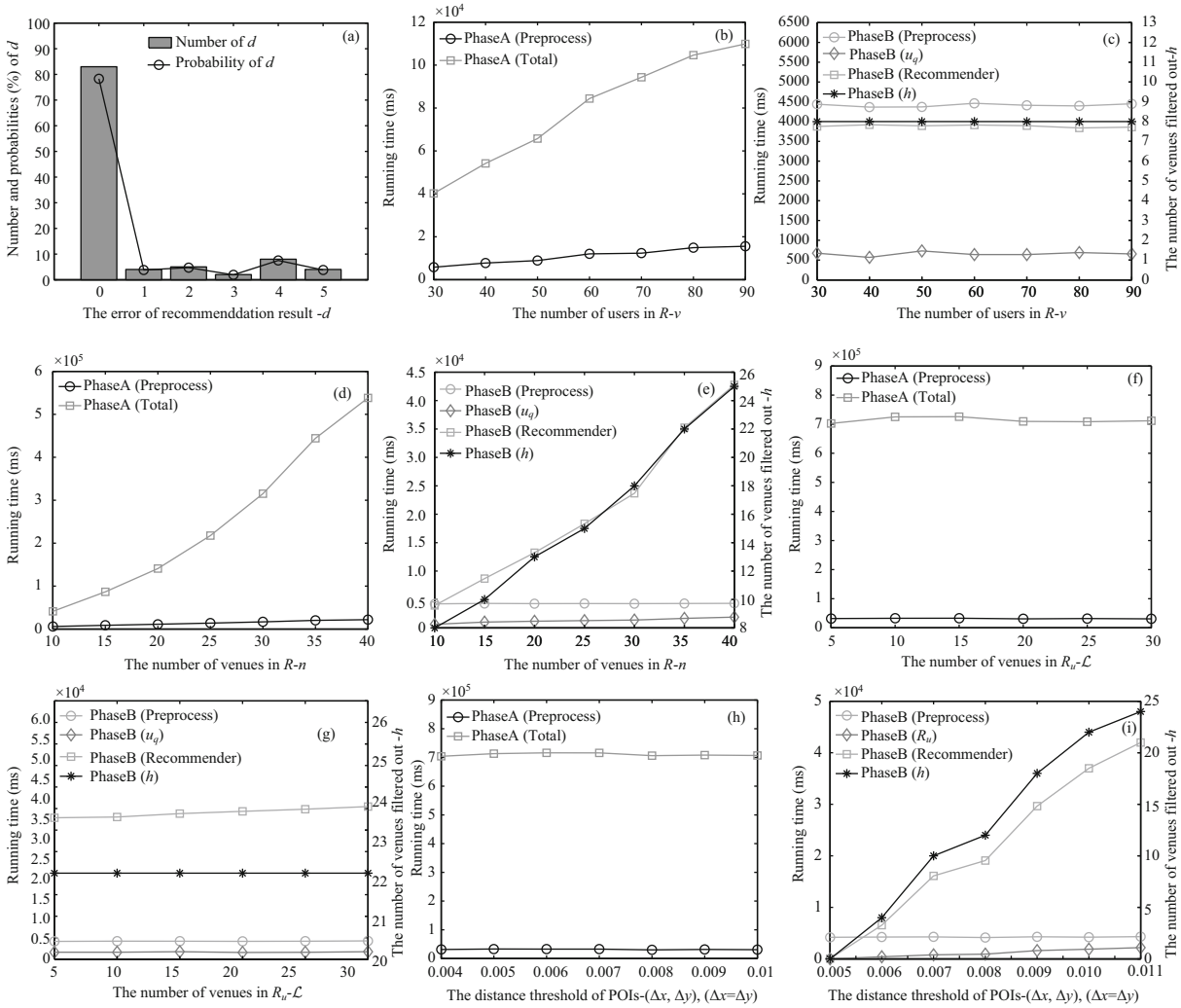

Figure 4 The effectiveness and efficiency of APPLET. (a) Recommendation quality of APPLET; (b) n=10, $\mathcal{L}$=5, $(\Delta x, \Delta y)$=(0.01, 0.01); (c) n=10, $\mathcal{L}$=5, $(\Delta x, \Delta y)$=(0.01, 0.01); (d) v=30, $\mathcal{L}$=5, $(\Delta x, \Delta y)$=(0.01, 0.01); (e) v=30, $\mathcal{L}$=5, $(\Delta x, \Delta y)$=(0.01, 0.01); (f) v=50, n=35, $(\Delta x, \Delta y)$=(0.01, 0.01); (g) $v = 50, n = 35, (\Delta x, \Delta y) = (0.01, 0.01)$; (h) v=50, n=35, $\mathcal{L}$=15; (i) v=50, n=35, $\mathcal{L}$=15.

reason for this is that the change in $(\Delta x, \Delta y)$ will lead to that of h. Therefore, CP and SP have to spend more time to recommender in Phase B. Moreover, u_q also needs more time to decrypt the ciphertext as h increases. However, the increase in the decryption time of u_q is also reasonable when varying $(\Delta x, \Delta y)$.

In conclusion, the results show that APPLET is sufficiently efficient to answer the request of u_q (in seconds). In addition, the privacy-preserving scheme does not bring about an overburden computation for the whole recommendation system.

5.3 Discussion with other schemes

In this section, we discuss the efficiency of APPLET along with two other state-of-the-art privacy-preserving recommendation systems. The first is a privacy-preserving personalized tweet recommendation framework (pTwitterRec) [20]. The other is a privacy-preserving friend recommendation (PPFR) system for online social networks [21]. Both systems adopt several encryption techniques to protect the user's privacy. Because these schemes are implemented under completely different settings (e.g., the number of clouds), they cannot be applied to our particular problem. Hence, we only discuss their efficiency, especially the response time, for u_q, because the user experience is the most important aspect of a recommender system. To be fair, we compared the efficiency using the same amount of data (i.e., 1000

items, and 30 historical ratings by u_q).

The pTwitterRec framework [20] was tested on a smartphone with a 1.512 GHz quad-core Krait CPU and 2 GB of RAM, and the results showed a computational overhead for users of 84.0911 s. PPFR [21] was implemented in C and tested on an Intel Xeon 6-Core 3.07 GHz PC with 12 GB of memory. The results show that users will require 3.57 s to compute a recommendation over 1000 items. In comparison, we also tested APPLET on the same sized dataset, the results of which indicate that users of ordinary Android devices only have to spend 10.96 s to receive a recommendation. Therefore, APPLET is sufficiently efficient to answer requests by u_q.

6 Related work

In the past few years, POI recommendation, also referred to as location-aware recommendation, has been recognized as an essential application in recommender systems [22]. For instance, by taking into account the locations of users and the items they like, Foursquare provides recommendations of places around the current location of the user. However, as the location privacy of users becomes more important, traditional location-aware recommender systems are facing a significant challenge, namely, how to protect the location privacy of users while preserving the recommendation quality. In this section, we review the state-of-the-art researches on location privacy and recommender systems.

Location privacy. There are several studies that have achieved location privacy, which are based on anonymity, differential privacy, and encryption schemes. The authors of [23–25] proposed some location-privacy preserving mechanisms (LPPMs) based on anonymity to protect the user's location privacy. Although these anonymity mechanisms are diversiform, each of them assumes the adversaries own specific prior knowledge. To solve the shortcomings of the above schemes, the authors of [26–28] introduced differential privacy mechanisms to protect the user's exact location independently from any side information that the adversary might possess. In addition, Shao et al. [29] proposed a fine-grained privacy-preserving LBS framework based on encryption, called FINE, for mobile devices. Notably, none of the work above can be directly used to protect the privacy in a recommender system, which also includes some other sensitive information. As a general encryption framework for SQL queries, CryptDB [30] can be used to query the ranges of positions in ciphertext using OPE. However, it cannot be used to implement a privacy-preserving recommendation because CryptDB only supports additional homomorphic encryption using Paillier encryption. However, during the recommendation process, we must multiply the ratings first and then sum the products to compute the similarities. Assuming that CryptDB is adopted to achieve the same purpose, we must query the database to obtain the plaintext first, and then implement the recommendation based on the plaintext. It is clear that this cannot achieve the security goals described in Subsection 2.4. Thus, CryptDB cannot be used to achieve the purpose of our scheme.

Recommender system. Some work (e.g., [31, 32]) has shown that a recommender system may obtain user privacy during a recommendation. In addition, Staff et al. [33] indicated that one key challenge was in balancing privacy, utility, and the overhead for end users when designing recommender systems. Thus, many researchers have devoted their efforts to studying a privacy-preserving recommender system. In [5, 34], they presented two privacy-preserving solutions based on anonymity and obfuscation techniques. In addition, Refs. [35–37] proposed some strong and formal privacy-preserving mechanisms based on differential privacy to protect user's privacy during a recommendation. Moreover, Refs. [20, 21, 38, 39] also introduced cryptology to protect user privacy in recommender systems. In addition, Guo et al. [40] proposed a trust-based fine-grained privacy-preserving friend recommendation scheme for OSNs. Xin et al. [41] explored a two-tiered notion of privacy, including a small set of "public" users and a large set of "private" users. Ma et al. [42] revised the user-based collaborative filtering technique, and proposed two privacy-preserving recommendation approaches fusing user-generated tags and social relations in a novel way. Aïmeur et al. [43] presented a privacy-preserving hybrid recommender system, consisting of serval different recommender algorithms. In [44], the authors also designed a mobile APP recommender system that considered the APPs' popularity and the security preference of users. However,

none of these researchers have designed a specialized privacy-preserving mechanism for a location-aware recommender system. Moreover, their methods also suffer from the inaccurate recommendation quality or low efficiency. In comparison, our APPLET perfectly protects the user's privacy by utilizing multiple encryption techniques while providing a high-quality recommendation.

7 Conclusion

The disclosure of user preferences in a recommender system seriously threatens users' personal privacy, especially when service providers move their user data to an untrusted cloud. In this paper, we presented a novel solution, called APPLET, to address the significant challenges in privacy-preserving location-aware recommender systems. For APPLET, we introduced multiple cryptography methodologies for protecting the privacy of the RUs without affecting the recommendation quality. Moreover, we evaluated the effectiveness and performance of APPLET, the results of which indicate that APPLET is an effective and efficient solution. As part of our future work, we may extend the framework to more complex recommendation services.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61202179, U1405255, 61502368, U1509214, U1135002), National High Technology Research and Development Program (863 Program) (Grant No. 2015AA016007), Shaanxi Provincial Natural Science Foundation (Grant No. 2015JQ6227), China 111 Project (Grant No. B16037), and Fundamental Research Funds for the Central Universities (Grant Nos. JB150308, JB150309).

Conflict of interest The authors declare that they have no conflict of interest.

References

1 Zheng Y, Capra L, Wolfson O, et al. Urban computing: concepts, methodologies, and applications. ACM Trans Intell Syst Tech, 2014, 5: 38
2 Sarwat M, Levandoski J J, Eldawy A, et al. LARS*: an efficient and scalable location-aware recommender system. IEEE Trans Knowl Data Eng, 2014, 26: 1384–1399
3 Brodkin J. Netflix shuts down its last data center, but it still runs a big it operation. http://arstechnica.com/information-technology/2015/08/netflix-shuts-down-its-last-data-center-but-still-runs-a-big-it-operation. 2015
4 Levi A, Mokryn O, Diot C, et al. Finding a needle in a haystack of reviews: cold start context-based hotel recommender system. In: Proceedings of the 6th ACM Conference on Recommender Systems, Dublin, 2012. 115–122
5 Celdran A H, Perez M G, Garcia C F, et al. PRECISE: privacy-aware recommender based on context information for cloud service environments. IEEE Commun Mag, 2014, 52: 90–96
6 Huang J, Qi J Z, Xu Y B, et al. A privacy-enhancing model for location-based personalized recommendations. Distrib Parallel Dat, 2015, 33: 253–276
7 Scipioni M P. Towards privacy-aware location-based recommender systems. In: Proceedings of the 7th International Federation for Information Processing Summer School, Trento, 2011. 1–8
8 Paillier P. Public-key cryptosystems based on composite degree residuosity classes. In: Advances in Cryptology — EUROCRYPT. Berlin: Springer, 1999. 223–238
9 Furukawa J. Request-based comparable encryption. In: Computer Security — ESORICS. Berlin: Springer, 2013. 129–146
10 Sarwar B, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th International Conference on World Wide Web, Hong Kong, 2001. 285–295
11 Dai W. Commutative-like encryption: a new characterization of ElGamal. arXiv:1011.3718
12 ElGamal T. A public key cryptosystem and a signature scheme based on discrete logarithms. In: Advances in Cryptology. Berlin: Springer, 1984. 10–18
13 Weis S A. New foundations for efficient authentication, commutative cryptography, and private disjointness testing. Dissertation for Ph.D. Degree. Cambridge: Massachusetts Institute of Technology, 2006
14 Furukawa J. Short comparable encryption. In: Cryptology and Network Security. Berlin: Springer, 2014. 337–352
15 Lu R X, Zhu H, Liu X M, et al. Toward efficient and privacy-preserving computing in big data era. IEEE Netw, 2014, 28: 46–50
16 Goldreich O. Foundations of Cryptography: Volume 2, Basic Applications. Cambridge: Cambridge University Press, 2009
17 Bost R, Popa R A, Tu S, et al. Machine learning classification over encrypted data. IACR Cryptology ePrint Archive, 2014, 331
18 Scott J. UMN/Sarwat foursquare dataset. https://archive.org/details/201309_foursquare_dataset_umn
19 Ye M, Yin P F, Lee W C. Location recommendation for location-based social networks. In: Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems, San Jose, 2010. 458–461

20 Liu B S, Hengartner U. pTwitterRec: a privacy-preserving personalized tweet recommendation framework. In: Proceedings of the 9th ACM Symposium on Information, Computer and Communications Security, Kyoto, 2014. 365–376
21 Samanthula B K, Cen L, Jiang W, et al. Privacy-preserving and efficient friend re-commendation in online social networks. Trans Data Privacy, 2015, 8: 141–171
22 Gao H J, Tang J L, Hu X, et al. Content-aware point of interest recommendation on location-based social networks. In: Proceedings of the 29th AAAI Conference on Artificial Intelligence, Austin, 2015. 1721–1727
23 Gao S, Ma J F, Shi W S, et al. TrPF: a trajectory privacy-preserving framework for participatory sensing. IEEE Trans Inf Forensic Secur, 2013, 8: 874–887
24 Niu B, Li Q H, Zhu X Y, et al. Enhancing privacy through caching in location-based services. In: Proceedings of IEEE Conference on Computer Communications (INFOCOM), Kowloon, 2015. 1017–1025
25 Cicek A E, Nergiz M E, Saygin Y. Ensuring location diversity in privacy-preserving spatio-temporal data publishing. VLDB J, 2014, 23: 609–625
26 Andrés M E, Bordenabe N E, Chatzikokolakis K, et al. Geo-indistinguishability: differential privacy for location-based systems. In: Proceedings of the 20th ACM SIGSAC Conference on Computer & Communications Security. Berlin: Springer, 2013. 901–914
27 Xiao Y H, Xiong L. Protecting locations with differential privacy under temporal correlations. In: Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, 2015. 1298–1309
28 To H, Ghinita G, Shahabi C. A framework for protecting worker location privacy in spatial crowdsourcing. Proc VLDB Endowment, 2014, 7: 919–930
29 Shao J, Lu R X, Lin X D. FINE: a fine-grained privacy-preserving location-based service framework for mobile devices. In: Proceedings of IEEE Conference on Computer Communications (INFOCOM), Toronto, 2014. 244–252
30 Popa R A, Redfield C, Zeldovich N, et al. CryptDB: processing queries on an encrypted database. Commun ACM, 2012, 55: 103–111
31 Calandrino J A, Kilzer A, Narayanan A, et al. "You might also like:" privacy risks of collaborative filtering. In: Proceedings of IEEE Symposium on Security and Privacy (S&P), California, 2011. 231–246
32 Bhagat S, Weinsberg U, Ioannidis S, et al. Recommending with an agenda: active learning of private attributes using matrix factorization. In: Proceedings of the 8th ACM Conference on Recommender Systems. New York: ACM, 2014. 65–72
33 Staff C. Recommendation algorithms, online privacy, and more. Commun ACM, 2009, 52: 10–11
34 Zhu J M, He P J, Zheng Z B, et al. A privacy-preserving QoS prediction framework for web service recommendation. In: Proceedings of IEEE International Conference on Web Services, New York, 2015. 241–248
35 Jorgensen Z, Yu T. A privacy-preserving framework for personalized, social recommendations. In: Proceedings of the 17th International Conference on Extending Database Technology, Athens, 2014. 571–582
36 Guerraoui R, Kermarrec A M, Patra R, et al. D2P: distance-based differential privacy in recommenders. Proc VLDB Endowment, 2015, 8: 862–873
37 Shen Y L, Jin H X. Privacy-preserving personalized recommendation: an instance-based approach via differential privacy. In: Proceedings of IEEE International Conference on Data Mining, Shenzhen, 2014. 540–549
38 Gong Y M, Guo Y X, Fang Y G. A privacy-preserving task recommendation framework for mobile crowdsourcing. In: Proceedings of IEEE Global Communications Conference, Austin, 2014. 588–593
39 Hoens T R, Blanton M, Steele A, et al. Reliable medical recommendation systems with patient privacy. ACM Trans Intell Syst Tech, 2013, 4: 67
40 Guo L, Zhang C, Fang Y G. A trust-based privacy-preserving friend recommendation scheme for online social networks. IEEE Trans Depend Secure Comput, 2015, 12: 413–427
41 Xin Y, Jaakkola T. Controlling privacy in recommender systems. In: Advances in Neural Information Processing Systems, Montreal, 2014. 3: 2618–2626
42 Ma T H, Zhou J J, Tang M L, et al. Social network and tag sources based augmenting collaborative recommender system. IEICE Trans Inf Syst, 2015, 98: 902–910
43 Aïmeur E, Brassard G, Fernandez J M, et al. Alambic: a privacy-preserving recommender system for electronic commerce. Int J Inf Secur, 2008, 7: 307–334
44 Zhu H S, Xiong H, Ge Y, et al. Mobile app recommendations with security and privacy awareness. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, New York, 2014. 951–960

Appendix A Proof of Theorem 1

Phase A. Construct a simulator S^1_{SP} that can simulate a view indistinguishable from the real view of SP, $V^{\pi 1}_{SP}(R, PK_s, SK_s, k_{1,2}, p; s, \alpha, C, W, \text{coins}; B)$ (where coins is a random tape for Paillier encryptions). Here, S^1_{SP} conducts as follows:

(1) Pick random numbers $\widetilde{s}$, $\widetilde{\alpha}$, and an $n \times m$ random matrix $\widetilde{C}$.

(2) Generate an $n \times m$ random Paillier encryption matrix: $\widetilde{B}$.

(3) Generate a random number matrix: $\widetilde{W}$.

(4) Generate a random tape for $n \times m$ Paillier encryptions: $\widetilde{\text{coins}}$.

(5) Output: $(R, PK_s, SK_s, k_{1,2}, p; \widetilde{s}, \widetilde{\alpha}, \widetilde{C}, \widetilde{W}, \widetilde{\text{coins}}; \widetilde{B})$.

We define the following hybrids:

- $H_0 = V^{\pi 1}_{SP}(R, PK_s, SK_s, k_{1,2}, p)$,
- $H_1 = (R, PK_s, SK_s, k_{1,2}, p; \widetilde{s}, \widetilde{\alpha}, \widetilde{C}, \widetilde{W}, \widetilde{\text{coins}}; B)$,
- $H_2 = S^1_{SP}(R, PK_s, SK_s, k_{1,2}, p, F)$.

Given that $(\widetilde{s}, \widetilde{\alpha}, \widetilde{C}, \widetilde{W}, \widetilde{\text{coins}})$ are generated according to the same distribution as $(s, \alpha, C, W, \text{coins})$, and that Paillier encryption is semantic secure, the hybrids $H_0 \overset{c}{\equiv} H_1$. Similarly, the distribution of $(\widetilde{B}, \widetilde{W})$ and (B, W) are exactly the same,

and the Paillier encryption is secure, and thus $H_1 \stackrel{c}{\equiv} H_2$. Hence, we showed that $V_{SP}^{\pi 1} \stackrel{c}{\equiv} S_{SP}^1$.

Then, a simulator S_{CP}^1 is constructed that can simulate a view indistinguishable from CP's real view $V_{CP}^{\pi 1}(E_{PK_s}(R), A; F)$. Here, S_{CP}^1 conducts as follows:

(1) Generate a $n \times m$ random Paillier encryption matrix $\widetilde{E_{PK_s}}(R)$ and a $n \times m$ random large number matrix $\widetilde{A}$.

(2) Generate a $n \times n$ matrix $\widetilde{F}$.

(3) Output: $(\widetilde{E_{PK_s}}(R), \widetilde{A}; \widetilde{F})$.

We define the hybrids $H_0 = V_{CP}^{\pi 1}(E_{PK_s}(R), A)$ and $H_1 = (\widetilde{E_{PK_s}}(R), \widetilde{A}; F)$.

Since Paillier encryption is semantic secure, we have $(E_{PK_s}(R), A) \stackrel{c}{\equiv} (\widetilde{E_{PK_s}}(R), \widetilde{A})$ and $H_0 \stackrel{c}{\equiv} H_1$. Hence, $V_{CP}^{\pi 1} \stackrel{c}{\equiv} S_{CP}^1$.

In addition, we encrypt A_v and $\{v_L\}$ through commutative encryption and comparable encryption, respectively. Given that these encryption schemes are secure, no probabilistic polynomial-time (P.P.T.) adversary can distinguish them. Therefore, we claim that no P.P.T. adversary can obtain SP's property and that Phase A in our APPLET is secure.

Phase B. Construct a simulator S_{u_q} that can simulate a view indistinguishable from u_q's real view $V_{u_q}^{\pi 2}((x_u, y_u), (\Delta x, \Delta y), sk_u, \text{param}, \text{mkey}, \alpha, k_1, p; \beta, \gamma, s', \text{coins}; E_{pk_u}(v'_N, v'_L), R'_p)$ (where coins is the random tape for comparable encryption). Here, S_{u_q} conducts as follows:

(1) Pick random numbers $\widetilde{\beta}, \widetilde{\gamma}$, and $\widetilde{s'}$.

(2) Generate an El-Gamal encryption vector $\widetilde{E_{pk_u}}(v'_N, v'_L)$, a random vector $\widetilde{R'_p}$ and a random tape for comparable encryption $\widetilde{\text{coins}}$.

(3) Output: $((x_u, y_u), (\Delta x, \Delta y), sk_u, \text{param}, \text{mkey}, \alpha, k_1, p; \widetilde{\beta}, \widetilde{\gamma}, \widetilde{s'}, \widetilde{\text{coins}}; \widetilde{E_{pk_u}}(v'_N, v'_L), \widetilde{R'_p})$.

We define the following hybrids:

- $H_0 = V_{u_q}^{\pi 2}((x_u, y_u), (\Delta x, \Delta y), sk_u, \text{param}, \text{mkey}, \alpha, k_1, p)$,
- $H_1 = ((x_u, y_u), (\Delta x, \Delta y), sk_u, \text{param}, \text{mkey}, k_1, p; \widetilde{\beta}, \widetilde{\gamma}, \widetilde{s'}, \widetilde{\text{coins}}; E_{pk_u}(v'_N, v'_L), R'_p)$,
- $H_2 = S_{u_q}((x_u, y_u), (\Delta x, \Delta y), sk_u, \text{param}, \text{mkey}, \alpha, k_1, p; (v'_N, v'_L), R_p)$.

Given that $(\widetilde{\beta}, \widetilde{\gamma}, \widetilde{s'}, \widetilde{\text{coins}})$ are generated with the same distribution as $(\beta, \gamma, s', \text{coins})$, and the comparable encryption is secure, the hybrids $H_0 \stackrel{c}{\equiv} H_1$. Because commutative encryption is secure and $(\widetilde{E_{pk_u}}(v'_N, v'_L), \widetilde{R'_p})$ follows the same distribution as $(E_{pk_u}\ (v'_N, v'_L), R'_p))$, we have $H_1 \stackrel{c}{\equiv} H_2$. Thus, $V_{u_q}^{\pi 2} \stackrel{c}{\equiv} S_{u_q}$.

Next, we construct a simulator S_{CP}^2 that can simulate a view indistinguishable from CP's view $V_{CP}^{\pi 2}(F, E_{PK_s}(R), pk_u, E_{pk_s}(v_N, v_L), \text{Enc}(v_L), \text{param}, k_2; Z, \text{coins}; \text{Enc}(\{x_u \pm \Delta x, y_u \pm \Delta y\}, \text{Der}(\{x_u \pm \Delta x, y_u \pm \Delta y\}), s', \beta^2, \alpha^2\gamma))$. Here, S_{CP}^2 conducts as follows:

(1) Generate a random area encrypted through comparable encryption: $\widetilde{\text{Enc}}(\{x_u \pm \Delta x, y_u \pm \Delta y\}), \widetilde{\text{Der}}(\{x_u \pm \Delta x, y_u \pm \Delta y\})$, a random matrix $\widetilde{Z}$, and random parameters: $\widetilde{s'}, \widetilde{\beta^2}, \widetilde{\alpha^2\gamma}$.

(2) Run the protocol using $\widetilde{\text{Enc}}(\{x_u \pm \Delta x, y_u \pm \Delta y\})$, $\widetilde{\text{Der}}(\{x_u \pm \Delta x, y_u \pm \Delta y\})$, $\widetilde{Z}$ and $\widetilde{s'}, \widetilde{\beta^2}, \widetilde{\alpha^2\gamma}$ as input.

(3) Generate a random tape for re-encryption in commutative encryption: $\widetilde{\text{coins}}$.

(4) Output: $(F, E_{PK_s}(R), pk_u, E_{pk_s}(v_N, v_L), \text{Enc}(v_L), \text{param}, k_2; \widetilde{Z}, \widetilde{\text{coins}}; \widetilde{\text{Enc}}(\{x_u \pm \Delta x, y_u \pm \Delta y\}), \widetilde{\text{Der}}(\{x_u \pm \Delta x, y_u \pm \Delta y\}), \widetilde{s'}, \widetilde{\beta^2}, \widetilde{\alpha^2\gamma})$.

We define the following hybrids:

- $H_0 = V_{CP}^{\pi 2}(F, E_{PK_s}(R), pk_u, E_{pk_s}(v_N, v_L), \text{Enc}(v_L), \text{param}, k_2)$,
- $H_1 = (F, E_{PK_s}(R), pk_u, E_{pk_s}(v_N, v_L), \text{Enc}(v_L), \text{param}, k_2; \widetilde{Z}, \widetilde{\text{coins}}; \text{Enc}(\{x_u \pm \Delta x,\ y_u \pm \Delta y\}), \text{Der}(\{x_u \pm \Delta x, y_u \pm \Delta y\}), s', \beta^2, \alpha^2\gamma)$,
- $H_2 = (F, E_{PK_s}(R), pk_u, E_{pk_s}(v_N, v_L), \text{Enc}(v_L), \text{param}, k_2; \widetilde{Z}, \widetilde{\text{coins}}; \widetilde{\text{Enc}}(\{x_u \pm \Delta x,\ y_u \pm \Delta y\}), \widetilde{\text{Der}}(\{x_u \pm \Delta x, y_u \pm \Delta y\}), \widetilde{s'}, \widetilde{\beta^2}, \widetilde{\alpha^2\gamma})$.

Given $\widetilde{Z}$ and $\widetilde{\text{coins}}$ generated as the distribution of Z and coins, we have $H_0 = H_1$. Through the security of comparable encryption and the same distribution of $\{\widetilde{s'}, \widetilde{\beta^2}, \widetilde{\alpha^2\gamma}\}$ with $\{s', \beta^2, \alpha^2\gamma\}$, we have $H_1 \stackrel{c}{\equiv} H_2$ and $H_2 \stackrel{c}{\equiv} S_{CP}^2$. Hence, $V_{CP}^{\pi 2} \stackrel{c}{\equiv} S_{CP}^2$.

Finally, we construct a simulator S_{SP}^2 that can simulate a view indistinguishable from SP's view $V_{SP}^{\pi 2}(sk_s, s^{-1} \bmod p, \alpha; E_{pk_u}(E_{pk_s}(v'_N, v'_L)), T, Q)$. Here, S_{SP}^2 conducts as follows:

(1) Generate a random re-encrypted vector: $\widetilde{E_{pk_u}}(E_{pk_s}(v'_N, v'_L))$, a random Paillier encryption vector: $\widetilde{Q}$ and a random vector: $\widetilde{T}$.

(2) Run the protocol with $\widetilde{E_{pk_u}}(E_{pk_s}(v'_N, v'_L)), \widetilde{Q}$ and $\widetilde{T}$.

(3) Output: $(sk_s, s^{-1} \bmod p, \alpha; \widetilde{E_{pk_u}}(E_{pk_s}(v'_N, v'_L)), \widetilde{T}, \widetilde{Q})$.

Then, the hybrids are defined as follows:

- $H_0 = V_{SP}^{\pi 2}(sk_s, s^{-1} \bmod p, \alpha; E_{pk_u}(E_{pk_s}(v'_N, v'_L)), T, Q)$,
- $H_1 = V_{SP}^{\pi 2}(sk_s, s^{-1} \bmod p, \alpha; \widetilde{E_{pk_u}}(E_{pk_s}(v'_N, v'_L)), \widetilde{T}, \widetilde{Q})$.

Since the security of commutative and Paillier encryption and $\{\widetilde{E_{pk_u}}(E_{pk_s}(v'_N, v'_L)), \widetilde{Q}, \widetilde{T}\}$ are generated as the distribution of $\{E_{pk_u}(E_{pk_s}(v'_N, v'_L)), Q, T\}$, we obtain $H_0 \stackrel{c}{\equiv} H_1$ and $H_1 \stackrel{c}{\equiv} S_{SP}^2$. Hence, $V_{SP}^{\pi 2} \stackrel{c}{\equiv} S_{SP}^2$.

To summarize, no P.P.T. adversary can distinguish the simulators' views from their own real views. Hence, no P.P.T. adversary can obtain u_q's private input or encrypted recommendation results. Moreover, no P.P.T. adversary can obtain the similarities of venues.