Achieving Privacy-Preserving Group Recommendation with Local Differential Privacy and Random Transmission

Hanyi Wang,1,2 Kun He,1,2 Ben Niu,1 Lihua Yin,3 and Fenghua Li1,2

1Institute of Information Engineering, Chinese Academy of Sciences, China
2School of Cyber Security, University of Chinese Academy of Sciences, China
3Cyberspace Institute of Advanced Technology (CIAT), Guangzhou University, China

Correspondence should be addressed to Ben Niu; niuben@iie.ac.cn

Received 16 March 2020; Revised 5 June 2020; Accepted 31 July 2020; Published 5 September 2020

Copyright © 2020 Hanyi Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Group activities on social networks are increasing rapidly with the development of mobile devices and IoT terminals, creating a huge demand for group recommendation. However, group recommender systems are facing an important problem of privacy leakage on user’s historical data and preference. Existing solutions always pay attention to protect the historical data but ignore the privacy of preference. In this paper, we design a privacy-preserving group recommendation scheme, consisting of a personalized recommendation algorithm and a preference aggregation algorithm. With the carefully introduced local differential privacy (LDP), our personalized recommendation algorithm can protect user’s historical data in each specific group. We also propose an Intra-group transfer Privacy-preserving Preference Aggregation algorithm (IntPPA). IntPPA protects each group member’s personal preference against either the untrusted servers or other users. It could also defend long-term observation attack. We also conduct several experiments to measure the privacy-preserving effect and usability of our scheme with some closely related schemes. Experimental results on two datasets show the utility and privacy of our scheme and further illustrate its advantages.

1. Introduction

With the development of the social networks and the mobile devices like smartphone or IoT terminals [1, 2], recommender systems, which are used to recommend for each individual, play an important role in our daily life. Nowadays, more and more people tend to gather together to enjoy social activities, such as watching movie or hiking together, having dinner with friends, or using apps such as Meetup or Douban to find out group activities with common-interest-strangers. With the latest development and popularity of smart devices and social networking services, it is convenient for people to form a group. Such trends bring new challenges to the existing recommender system: which items or events (e.g., movie/attraction/restaurant) should be recommended, in order to satisfy all/-most of the group members? Under this circumstance, group recommendation has gradually become one of the hotspots in the field of recommender systems [3–8].

However, the privacy issue is obstructing the development of recommender system. In 2018, about 87 million Facebook users’ information was leaked to Cambridge Analytica company. By utilizing such information, the company built user model and obtain users’ personal preferences. Based on these preferences, the company recommended targeted promotion content about the US election to these users. It affects users’ votes to some extent, causing serious violations of human rights. Therefore, we should pay much attention to protect users’ privacy on recommender systems, including both of their historical data and personal preference.

Many privacy-preserving approaches have been proposed to protect users’ historical data during recommendation over recent years. Yang et al. [9] proposed a framework called PrivRank. In PrivRank, both historical data and activity data of users are protected by obfuscation. In the mean time, these obfuscated data could provide high-quality personalized ranking-based recommendation services. In the
eHealthcare system, Xu et al. [10] proposed a privacy-preserving online medical service recommendation scheme. This scheme uses Paillier encryption algorithm to match each patient's needs with the information of doctors. It recommends appropriate doctors to patients, without knowing patients' exact demands.

Although these work protect users' historical data during recommendation, they ignore that results of personalized recommendation also leak user's privacy. For example, if an untrusted server learns a specific user's preference on restaurants, without knowing the exact restaurants that the user went to, the server could easily figure out user's food taste, places that he or she usually went, and how much he or she may cost on food. And the server probably infers user's salary or even home address according to these preferences.

For this concern, we take the privacy of personal preference into consideration. In that case, there are three challenges we need to solve. First, we need to hide both historical data and preferences of users from honest but curious parties during the recommendation. We also need to guarantee the accuracy of recommendation results at the same time. Second, since a group is dynamic, we should focus on the long-term observation attacks launched by the malicious users in the group recommendation. Third, how to evaluate our recommendation scheme is also a tricky issue.

The main contributions of our paper are summarized as follows:

(i) We propose a privacy-aware group recommendation scheme that protects each user's historical data and personal preferences at the same time. The scheme consists of two algorithms. The former focuses on the personalized recommendation problem, which guarantees ε-LDP for each user. The latter is a privacy-preserving preference aggregation algorithm called IntPPA, designed to solve the group recommendation problem. By adding noise on each member's preference profile, and transferring them within the group, the user's preferences can be protected, and the accuracy of the group recommendation is also guaranteed.

(ii) In IntPPA algorithm, no one could easily infer the members' preferences through long-term observation attacks. We also adopt a median aggregation strategy to prevent malicious group members from tampering data.

(iii) We conduct several experiments on two real-world datasets. We use RMSE and F-score to measure the utility, and use RMSE and "matched pairs" to measure the privacy-preserving effect under long-term observation attacks. The results show our scheme has good utility and privacy-preserving effect.

The rest of this paper is organized as follows. In Section 2, we review some recent work. Section 3 introduces the preliminaries. In Section 4, we elaborate the details of our proposed methods. We also demonstrate our evaluation results in Section 5. Finally, we conclude our paper in Section 6.

2. Related Work

2.1. Privacy-Preserving Personalized Recommendation. Differential privacy (DP) has been widely utilized in the personalized recommender systems to protect users' privacy. Under DP, the recommender systems could not infer a user's profile from the changes of recommendation results. McSherry and Mironov [11] proposed a differential privacy method based on collaborative filtering algorithm. They added noise not only on the sum, count, or average but also on the covariance matrix and then sent it to the recommender system. Even though the method could protect users' privacy, the performance is of low utility since too much noise has been added.

Then, Liu et al. [12] and Balu and Furon [13] also proposed recommendation algorithms based on DP and improved the performance. However, most of them assumed that the recommender system is a trusted third party.

Hua et al. [14] achieved differentially private matrix factorization based on a gradient descent algorithm, which prevents users' private ratings from the untrusted recommender. Shen et al. [15] proposed a differentially private framework without any trusted third parties. By solving optimization problems, the method could perturb users' historical records, and guarantee users' category preferences under ε-DP.

Some solutions utilized local differential privacy to construct privacy-preserving recommender systems. Shin et al. [16] proposed a personalized recommendation algorithm based on matrix factorization and used a LDP mechanism on the gradient of users' profiles. The algorithm not only protected users' ratings, but also the items that were rated by the users. Shin et al. [16] also enhanced their algorithm via dimension reduction in order to improve the accuracy and decrease the computing costs. Since this algorithm obtains recommendation results on the user side, only users can learn the recommended result.

2.2. Group Recommendation. Group recommendations try to satisfy a group of users' preferences. It could be applied in various areas such as video services [17], shopping [18], traveling [19], having dinner [20], and even in IoT scenario [21]. How to extract the common preferences of group members, reduce the preference conflict among group members, and make the recommendation results satisfy the needs of all group members as much as possible, are the key problems in group recommendation system.

Solutions to group recommendation are usually divided into two categories. One category aggregates the profiles of the group members into one profile, then regards it as a user profile, and makes personalized recommendations based on the aggregated profile [22, 23]. On the contrary, the other methods first make personalized recommendations for each group member, respectively, and then aggregates their recommendation results as the group recommendation result by aggregation strategies [24, 25]. Comparing with the former methods, the latter ones have better flexibility and also obtain more attentions.

A few work solve the privacy problems in group recommendation. Luo and Chen [26] proposed a group-based privacy-preserving method. The algorithm perturbed the
preference data of each group member and then utilized CF-based group recommendation to aggregate their data. Shang et al. [27] proposed a ranking-based privacy-preserving group recommendation method. Group members exchanged their obfuscated profiles to protect sensitive information. And they constructed a hybrid collaborative filtering model based on Markov random walks to provide recommendations and predictions to group members.

3. Preliminaries

3.1. Motivation and Basic Idea. There is a huge privacy risk in personalized recommendation. Once the attacker collects the user’s historical data in the recommendation, the user’s sensitive information could be inferred. For example, we often rate the restaurants we have visited in the Dianping or Yelp. According to these data, the attacker can easily infer the user’s historical geographical location and even further speculate the user’s financial conditions and home address.

On the other hand, with the development of social network, the interaction between users increases. Recommender systems are no longer limited to recommending items to one user, but gradually extended to group-oriented services.

Most group recommendation algorithms need to integrate the personalized recommendation results of each group members, so the risks in personalized recommendation also hold true for group recommendation. However, in addition to historical data, users’ personalized recommendation results are also very sensitive. Personalized recommendation results reflect users’ preferences and behavior habits, involving food preferences, entertainment interests, social network relationships, political tendencies, and other aspects, thus forming a “user portrait” of people.

Existing privacy-preserving algorithms of recommendation system only protect user’s historical data but ignore the disclosure risk of personalized recommendation results. Moreover, a group tends to change and update, so the attacker is likely to analyze more user information through the long-term observation. How to further protect the personalized recommendation results and resist the long-term observation attack in group recommendation is the main problem to be solved in this paper.

We design a privacy-aware group recommendation scheme. Figure 1 further illustrates our basic idea, which could be divided into two parts. Steps 1, 2, and 3 show the personalized recommendation part. Users interact with the recommendation server under local differential privacy, in order to train the personalized recommendation model via matrix factorization mechanism without privacy leakage, and then the personalized recommendation results will be generated on the client side. After the personalized recommendation part, each user obtains his or her own preferences. Steps 4 and 5 represent the preference aggregation part. When a group needs a recommendation, the group will execute the InPPA algorithm. Each group member’s personal preference is transferred inside the group with perturbation before sending to the server. In this case, the real preferences of group members are hidden during the random transmission and will not be exposed to others. After receiving all members’ perturbed preferences, the server adopts the median aggregation strategy to fuse them and obtains the final group preference.

3.2. Local Differential Privacy. Local differential privacy is a state-of-the-art notion that guarantees users to share private data safely.

Definition 1. A privacy mechanism $\mathcal{M}$ satisfies $\varepsilon$-local differential privacy ($\varepsilon \geq 0$) if for any two different records $t, t' \in \text{Domain}(\mathcal{M})$, and for any output $t^* \in \text{Range}(\mathcal{M})$, 

$$\Pr(\mathcal{M}(t) = t^*) \leq e^\varepsilon \Pr(\mathcal{M}(t') = t^*).$$  \hspace{1cm} (1)

Local differential privacy has a sequential composibility property same as differential privacy. Sequential compositibility means the privacy budget $\varepsilon$ could be assigned in different steps of an algorithm.

![Figure 1: Framework of our proposed scheme.](image-url)
Property 2. For $n$ privacy-preserving algorithms $M_i$, $1 \leq i \leq n$. If algorithm $M_i$ satisfies $\varepsilon_i$-local differential privacy, algorithm $M = \{M_1, \ldots, M_n\}$ satisfies $\varepsilon$-LDP, which $\varepsilon = \sum_{i=1}^{n} \varepsilon_i$.

3.3. Problem Statement. We take the movie recommender system as an example to introduce our scheme in this paper. We assume that there is a movie recommender system consisting of $m$ movies and $n$ users. We denote the rating generated by user $i$ for movie $j$ as $b_{ij}$. If user $i$ has not rated movie $j$, $b_{ij} = \text{null}$. The movie recommender system has many groups that are formed by users. We denote a typical group with $g$ users as $G$. The notations are listed in Table 1.

In a group recommender system, it first collects users’ historical data $b_i$ to predict preference rating $b^*_i$ for movies that are not rated by users and then aggregates $b^*_{ij}$ (for all $i \in G$) for each movie $j$ to obtain group preference rating $gr_j$ for movie $j$. Usually, the sensitive historical ratings and preference ratings will be transmitted between users and server without protection, which increases the risk of users’ privacy leakage. In our work, we assume that the server is untrusted and each user only trusts themselves. And the adversaries are assumed to have access to the output data of all users and know the privacy-preserving scheme adopted by the users. Since groups tend to change and update, attackers also try to find out the connection between users’ identities and their sensitive data by carrying out the long-term observation attacks. In these cases, how to build a group recommender system without privacy leakage is our main problem.

We are facing several challenges to solve the problem. First, how to protect users’ personal preference ratings $b_{ij}$ from leakage to others except the rating owner and guarantee the accuracy of group recommendation results at the meantime? Second, how to defend against long-term observation attacks? Third, how to measure the privacy-preserving effect against long-term observation attack?

4. Our Proposed Scheme

4.1. Personalized Recommendation under Local Differential Privacy Algorithm. To fully protect users’ personal profiles, the privacy-preserving algorithm should not only prevent users’ historical data from leakage but also need to pay attention to their personalized recommendation results, during the personalized recommendation step. So based on the algorithm proposed by Shin et al. [16], we refine the matrix factorization method to fit in with our group recommendation scenario.

To solve the matrix factorization problem without over-fitting, we minimize the following formula:

$$\sum_{i,j} y_{ij}(b_{ij} - u_i v_j)^2 / N + \lambda \left( \sum_{i=1}^{n} ||u_i||^2 + \sum_{j=1}^{m} ||v_j||^2 \right).$$  \hspace{1cm} (2)

In formula (2), user vector $u_i = (u_{i,1}, u_{i,2}, \ldots, u_{i,d})$ represents user $i$’s relations with $d$ latent factors separately. These factors also have inter-dependencies with movies, which are denoted as item vectors $v_j = (v_{j,1}, v_{j,2}, \ldots, v_{j,d})^T$ for movie $j$.

Matrix $Y = \{y_{ij}\}_{i,j}^{n,m}$ represents whether a user has given a rate to a movie. If $b_{ij} = \text{null}$, $y_{ij} = 0$, else $y_{ij} = 1$. $N = \sum_{j=1}^{m} y_{ij}$, which is the total number of the ratings in the system.

We utilize a privacy-preserving gradient descent algorithm to approach the minimum of formula (2).

Users separately interact with the server for $k$ times to achieve the gradient descent algorithm. According to Property 2, if our algorithm needs to realize $\varepsilon$-LDP, then each iteration needs to realize $\varepsilon/k$-LDP. Before each iteration, users and the server need to initialize $u_i$ and $V$ on their own sides. During each iteration, $V$ is sent to each user first, then user $i$ computes a $d \times m$ matrix $X_i$, where each element $(x_i)_{lj} = -2 y_{ij} u_{lj}(b_{ij} - u_i v_j)$. It is relative to the gradient of formula (2) on user vectors. According to Shin’s work, each $(x_i)_{lj}$ needs to be normalized into range $[0, 1]$. The min-max normalization is usually used to normalize, but it will leak users’ gradient range. So, we propose a normalization method without too much information leakage. Each user $i$ computes $\text{Max}_i = \max(\{|(x_i)_{0,j}|, \ldots, |(x_i)_{d,m}|\})$, then $(x_i)_{lj} = (x_i)_{lj}/\text{Max}_i$.

Then, each user uses LDP algorithm to perturb $(x_i)_{lj}$. User $i$ randomly chooses two elements $l, j \in [1, d], j \in [1, m]$, and computes $c_i = (x_i)_{lj}$.
Next, user \( i \) perturbs \( c_i \) in the following distribution and obtains \( c'_i \). We denote \( t(\varepsilon) = \text{mdLog}(e^{\varepsilon k} + 1)/(e^{\varepsilon k} - 1) \):

\[
\Pr \left( c'_i = c_0 \right) = \begin{cases} 
\frac{c'(e^{\varepsilon k} - 1)}{2(e^{\varepsilon k} + 1)} + \frac{1}{2} & \text{when } c_0 = t(\varepsilon), \\
\frac{-c'_i(e^{\varepsilon k} - 1)}{2(e^{\varepsilon k} + 1)} + \frac{1}{2} & \text{when } c_0 = -t(\varepsilon).
\end{cases}
\]

(3)

After that, each user \( i \) sends \( c'_i \), \( l_i \), \( j_i \) to the server, then updates user vector \( u_i = u_i - \gamma[-2/N\sum_j y_{ij}v_j^T(h_{ij} - u_j, v_j) + 2\lambda u_j] \), where \( \gamma \) is the learning rate of the gradient descent algorithm. After receiving information from all users, the server adds each \( c'_i/N \) to the row \( l_i \) and column \( j_i \) of matrix \( V_v = \{0\}_{d \times m} \), in order to estimate the gradients of \( V \). Then, server updates \( V = V - \gamma(V + 2\lambda V) \) and sends it back to all users.

Users and the server interact \( k \) times as above, and finally each user \( i \) computes \( b'_{ij} = u_i V \) on their own sides and obtains each items’ predicted ratings.

The following theorems illustrate the privacy and usability of this algorithm, which have been proved similarly in [16], so we are not going to prove them here.

**Theorem 3.** For each user, the algorithm in Section 4.1 satisfies \( \varepsilon \)-local differential privacy.

**Theorem 4.** There is \( E(V_{ij}) = V_{ij}' \) represents the true value of the item matrix under no privacy-preserving algorithm. \( V_{ij} \) represents the perturbed value of item matrix elements under this algorithm.

Theorem 3 indicates the functionality on the privacy protection of our algorithm, while Theorem 4 shows its utility, since \( V_{ij} \) is the unbiased estimation of \( V_{ij}' \).

### 4.2. IntPPA Algorithm

This algorithm corresponds to steps 4 and 5 in Figure 1. It consists of an inter-group random transmission and an aggregation strategy. We first introduce why we choose median strategy to be our aggregation strategy.

There are lots of aggregation strategies in group recommendation such as average strategy. Under this strategy, the average of group members’ ratings represents the group preference. Also, there is a typical strategy called least misery. Under this strategy, the minimum value of group members’ ratings represents the group preference.

Suppose a group consists of five members: Alice, Bob, Charles, David, and Eric. Each member has his or her own preferences on five movies (predicted via personalized recommendation), which is shown in Table 2. The server supposes to select two movies to satisfy most of the members. If the average strategy is chosen, the group preferences for these 5 movies will be 1, 2, 2, 1, and 1, separately. And Movies 2 and 3 will be recommended to the group. If the least misery strategy is chosen, the group preferences for these 5 movies will be 1, 2, 2, 1, and 1, separately, and the server will also recommend Movies 2 and 3. In this paper, we choose median strategy to measure the group preference. Under this strategy, the median value of members’ ratings represents the group preference. So, the group preference will be 3, 4, 4, 3, and 3, separately, and Movies 2 and 3 will be recommended. According to the definition, median strategy is less affected by extreme ratings. We assume there is a dishonest user Gloria, she is a new member of this group, and she wants the server to choose Movie 4, so she changed her predicted ratings into 1, 1, 1, 1. In this case, the group will recommend Movies 3 and 4 under average aggregation strategy. While under least misery strategy, all movies has the same preference, which is indistinguishable for the server, so this strategy is unavailable under this situation. For our median aggregation strategy, Gloria’s plan does not work out. Movies 2 and 3 still have the highest ratings, which means Movie 4 will not be recommended. Based on these analysis, it is hard for the malicious users in the group to affect the group preference by changing their predicted scores under median aggregation strategy.

Next, we are going to introduce the random transmission. Suppose there is a group \( G \), who has \( g \) group members. Group members aim to send profile \( b^* \) to the server without privacy leakage, and at the meantime, the server could obtain the group preference of each item. In order to prevent the server from knowing users’ personal preferences, we design a random transmission mechanism inside the group, called IntPPA.

Server sets two time parameters: \( t_{\text{start}} \) and \( t_{\text{end}} \). All the members have to start this IntPPA algorithm after time \( t_{\text{start}} \) and finish before time \( t_{\text{end}} \). We propose a perturbed algorithm \( pf(b^*_i) = (b_{i1}', f_1, b_{i2}', f_2, \ldots, b_{im}', f_m) \), and for each \( f_j \), \( Pr (f_j = \Delta f) = 0.5 \), \( Pr (f_j = -\Delta f) = 0.5 \). \( \Delta f \) is a fixed value, and we call it as “change value.”

Group members first perturb their profile \( b^* \) into \( b_i' = pf(b^*_i) \). Then, they send perturbed profiles to the server with the probability of \( 1 - p_0 \) and send to each group member with the probability of \( p_0/g \). When a member receives others’ profiles, he/she also needs to use the perturbed algorithm to update \( b_i' = pf(b_i') \) first and then sends \( b_i' \) to group members or server according to the probability distribution described above. A member’s profile will not stop to be transmitted until the profile is sent to the server or the time exceeds \( t_{\text{end}} \).

The server finally receives all \( b_i' \) and computes group recommendation results \( gr = (gr_1, gr_2, \ldots, gr_m) \), in which \( gr_j = \text{median}(b_{1j}', b_{2j}', \ldots, b_{mj}') \). \( gr_j \) represents the group

| Table 2: A rating matrix. |
|--------------------------|
| Movie 1 | Movie 2 | Movie 3 | Movie 4 | Movie 5 |
| Alice   | 5       | 4       | 5       | 3       | 5       |
| Bob     | 3       | 4       | 3       | 2       | 3       |
| Charles | 5       | 4       | 2       | 1       | 3       |
| David   | 2       | 3       | 5       | 3       | 1       |
| Eric    | 1       | 2       | 4       | 5       | 2       |
| Gloria  | 1       | 1       | 5       | 5       | 1       |
During the algorithm, user 1 perturbs the profile into \((a + \Delta f, b + \Delta f, c - \Delta f, \cdots)\) and transmits to user 2. Then, user 2 perturbs it into \((a, b + 2\Delta f, c, \cdots)\) and transmits to user 3. After that, user 3 transmits \((a - \Delta f, b + \Delta f, c + \Delta f, \cdots)\) to user 1, and then user 1 transmits \((a - 2\Delta f, b, c, \cdots)\) to user 4. Finally, user 4 submits \((a - \Delta f, b + \Delta f, c - \Delta f, \cdots)\) to the server.

Under IntPPA algorithm, neither the server nor users could recognize whom the profiles are exactly from. When a new group member joins in this group, the server also cannot find out the fresh member’s profile, according to the difference between results computed before and after the member’s attendance.

5. Performance Evaluation

5.1. Experimental Settings

5.1.1. Datasets. We evaluate our experiments on two datasets. The first dataset is MovieLens-100k [28], which contains one hundred thousand movie ratings from approximately 1000 users over 1500 movies. The second dataset is FilmTrust [29]. FilmTrust is a dataset crawled from the entire FilmTrust website. Table 3 shows the details of the two datasets.

5.1.2. Comparison Methods. In Section 2.2, we mentioned several privacy-preserving group recommendation work. However, we did not compare our scheme with theirs. Although their works protect users’ ratings and even could protect users’ preferences to some extent, they did not consider the long-term observation attack during the group recommendation. While we assume a stronger attack model, it is not appropriate to compare with them. In this case, we compare our scheme (priv-MF-IntPPA) with some baselines in our experiments: priv-MF-med, priv-MF-avg, and priv-MF-lm.

The difference between ours and the other three methods is the aggregation strategy. Priv-MF-med, priv-MF-avg, and priv-MF-lm choose median, average, and least misery aggregation strategies, respectively. They have the same privacy-preserving personalized recommendation methods as ours, while they do not apply privacy protection to the preference aggregation part.

5.1.3. Utility Metrics. We evaluate the data utility of personalized recommendation algorithm by computing the RMSE (root mean squared error) between the item scores predicted by the training set and the actual score of the test set. We evaluate the group recommendation accuracy performance using precision (Pre@\(G_0\)) and recall (Rec@\(G_0\)). Here, \(G_0\) is the number of the recommended items. We evaluate the experiment with \(G_0 = \{5, 10, 15\}\). Pre@\(G_0\) is the fraction of top \(G_0\) recommendations selected by the group, and Rec@\(G_0\) is the fraction of true items retrieved in the top \(G_0\) recommendations.

5.1.4. Privacy Metrics. We use two metrics to evaluate the privacy-preserving effect of IntPPA algorithm. We compute the RMSE between each user’s profile before and after the IntPPA algorithm. We also propose a “matched pair” metric to describe the privacy-preserving effect against long-term observation attack.

We execute the IntPPA algorithm on the same group for multiple times and compute the differences between the perturbed profiles. For example, if user \(i\)’s profile in execution one is closest to user \(i’\)’s in execution two, we will call them as “a pair.” If these two members have the same identity, which means user \(i\) is actually user \(i’\), we will call that pair “matched.” We denote the number of matched pairs in a group as \(mp\) and normalize it to interval \([0, 1]\). Obviously, the smaller \(mp\) is, the better the privacy protection performs against the long-term observation attack.

5.2. Evaluation Results. We follow the evaluation method of privacy protection proposed in Li et al. [30] to measure our scheme.

We first did some pilot experiments to screen appropriate parameters of the method. According to the pilot experiments, we set number of latent factors \(d = 10\) and number of iterations \(k = 10\). We also set \(\gamma = 6 \times 10^{-6}, \lambda = 2 \times 10^{-3}\) in MovieLens’ experiments and \(\gamma = 8 \times 10^{-6}, \lambda = 2 \times 10^{-3}\) in FilmTrust’s.
5.2.1. Personalized Recommendation Utility Analysis. We divide the dataset into five folds, using four folds as training set and the other as testing set. We use RMSE to measure the utility of personalized recommendation algorithm. In Figures 3(a) and 3(b), the dotted line displays the utility of non-privacy matrix factorization algorithm, while the red line displays the utility of ours. In both datasets, the red line approaches to the dotted line as the privacy budget $\varepsilon$ increases, which means $\varepsilon$ has a negative correlation with the utility of our personalized recommendation algorithm. As we know, $\varepsilon$ has a positive correlation with the privacy-preservation effect. So, in order to balance the privacy and utility, we set $\varepsilon = 0.4$ for both datasets in the following experiments.

5.2.2. IntPPA Algorithm Privacy-Preserving Effect Analysis. We utilize two methods to measure the privacy-preserving effect of IntPPA. Since these measurements are only relevant with parameter $\Delta f$, $p_0$, and $g$, we test it on MovieLens dataset.

Figure 4(a) shows that larger $\Delta f$ or larger probability $p_0$ achieves better performance on privacy protection. In terms of parameter selection, when the data in the dataset has wider range, we need to increase $\Delta f$ or $p_0$ to guarantee a suitable perturbation.

Second, we utilize “matched pairs” $mp$ to measure the privacy. Figure 4(b) shows that when transfer probability $p_0$ increases, “matched pairs” will decrease, which means a better performance on privacy protection. Change value $\Delta f$ and group size $g$ are also negatively correlated with $mp$. However, once $p_0$ is smaller than 0.1, the IntPPA algorithm could not resist long-term observation attack.

According to the above experiments, we choose parameter $\Delta f = 0.5$, $p_0 = 0.7$ for both MovieLens and FilmTrust datasets.

5.2.3. Group Recommendation Accuracy Analysis. Since both MovieLens and FilmTrust do not contain group information, we extract groups that are randomly chosen from the users to build groups.
We use \( \text{Pre}@G_0 \) and \( \text{Rec}@G_0 \) to evaluate our scheme’s recommendation accuracy. In MovieLens (FilmTrust), if a group member gives score 4 (3) or above to a movie, we assume that the movie is adopted by the member.

Figures 5 and 6 report the \( \text{Pre}@G_0 \) and \( \text{Rec}@G_0 \) values for the two datasets with \( G_0 = \{5,10,15\} \) and group size \( g = \{5,10\} \). We compare our scheme (priv-MF-IntPPA) with the other group recommendation algorithms (priv-MF-med, priv-MF-avg, and priv-MF-lm) described in Section 5.1. We observe from the two figures that

(i) For priv-MF-IntPPA, when \( G_0 \) becomes larger, \( \text{Pre}@G_0 \) has a trend to become smaller, but \( \text{Rec}@G_0 \) becomes larger.

(ii) Group size \( g \) has little effect on the recall and precision value of our group recommendation.

(iii) In the MovieLens dataset, the precision of our scheme is close to others’, and sometimes even be better than them due to the randomness. However, the recall of our scheme is a little weak. In Figure 5(c), when \( G_0 = 5 \), the recall values of priv-MF-IntPPA and priv-MF-med are 0.0021 and 0.0039. Priv-MF-med improves 85.7% over our scheme. But when \( G_0 = 15 \), the recall values are 0.0070 and 0.0086; priv-MF-med only improves 22.9% over our scheme. So, as group size becomes larger, the differences will be acceptable.

(iv) In the FilmTrust dataset, the precision and recall values of each method are much lower than the values in the MovieLens dataset. It is probably because FilmTrust has lower rating density. Figure 6(c) shows that our scheme performs much better than other methods, which means that our algorithm is more suitable for low sparsity data than other algorithms.

From the above, our scheme can provide accurate group recommendation under the premise of ensuring privacy.

5.3. Communication and Time Cost. As for personalized recommendation, we only analyze the communication cost in one iteration. For each user, no matter how many items the user rates, only three elements are transmitted to the server, which is about \( 24 \text{ B} \) for both MovieLens and FilmTrust datasets. During each iteration, server transmits a matrix \( V \) of \( dm \)
elements to each user, which are about 0.15 MB for MovieLens and 0.47 MB for FilmTrust.

In IntPPA algorithm, server has no communication cost. Users transmit elements to the server and group members, which are less than 1 MB during our algorithm in both datasets.

According to our test, under the MovieLens (FilmTrust) dataset, the time of each iteration in personalized recommendation is no more than 90 (180) seconds. The time consumed by other parts can be ignored.

6. Conclusions

In this paper, we proposed a privacy-aware group recommendation scheme, consisting of a personalized recommendation algorithm and a preference aggregation algorithm. For the personalized recommendation, we employed local differential privacy to protect user’s historical data and prevent predicted preferences from leakage. We also designed an intra-group transfer privacy-preserving preference aggregation algorithm called IntPPA. IntPPA could not only protect users’ privacy but also defend against long-term observation attacks. Moreover, we presented several experiments to measure the privacy effect and usability of our proposed scheme. The effect and efficiency of our proposed scheme on the MovieLens and FilmTrust datasets show our advantages.

Data Availability

Our datasets are from MovieLens-1M dataset. The MovieLens-1M data supporting our recommendation system are from previously reported studies and datasets, which have been cited [28].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (61932015, 61872441, 61872100), the National Key R&D Program of China (2017YFB0802203), and the Youth Innovation Promotion Association, Chinese Academy of Sciences (2018196).
References

[1] X. Chen, A. Li, X. Zeng, W. Guo, and G. Huang, "Runtime model based approach to iot application development," Frontiers of Computer Science, vol. 9, no. 4, pp. 540–553, 2015.

[2] Y. Yang, X. Zheng, and C. Tang, "Lightweight distributed secure data management system for health internet of things," Journal of Network and Computer Applications, vol. 89, pp. 26–37, 2017.

[3] S. A. Yahia, S. B. Roy, A. Chawlat, G. Das, and C. Yu, "Group recommendation: Semantics and efficiency," Proceedings of the VLDB Endowment, vol. 6, no. 1, pp. 754–765, 2009.

[4] L. A. M. C. Carvalho and H. T. Macedo, "Users' satisfaction in recommendation systems for groups: an approach based on noncooperative games," in Proceedings of the 22nd ACM International Conference on World Wide Web, pp. 951–958, 2013.

[5] V. Rakesh, W. C. Lee, and C. K. Reddy, "Probabilistic group recommendation model for crowdfunding domains," in Proceedings of the 9th ACM International Conference on Web Search and Data Mining, pp. 257–266, 2016.

[6] T. D. Q. Vinh, T. A. N. Pham, G. Cong, and X. L. Li, Attention-based group recommendation with virtual coordinators, Future Generation Computer Systems, vol. 94, pp. 224–236, 2019.

[7] Q. Yuan, G. Cong, and C. Y. Lin, "COM: A generative model for group recommendation," in Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 163–172, 2014.

[8] D. Yang, B. Qu, and P. Cudré-Mauroux, "Privacy-preserving social media data publishing for personalized ranking-based recommendation," IEEE Transactions on Knowledge and Data Engineering, vol. 31, no. 3, pp. 507–520, 2019.

[9] C. Xu, J. Wang, L. Zhu, C. Zhang, and K. Sharif, "Ppmr: a privacy-preserving online medical service recommendation scheme in ehealthcare system," IEEE Internet of Things Journal, vol. 6, no. 3, pp. 5665–5673, 2019.

[10] F. McSherry and I. Mironov, "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, 2009.

[11] Z. Liu, Y. Wang, and A. Smola, "Fast differentially private matrix factorization," in Proceedings of the 9th ACM Conference on Recommender Systems, pp. 171–178, 2015.

[12] R. Balu and T. Furon, "Differentially private matrix factorization using sketching techniques," in Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security, pp. 57–62, 2016.

[13] J. Hua, C. Xia, and S. Zhong, "Differentially private matrix factorization," in Proceedings of the 24th International Joint Conference on Artificial Intelligence, pp. 1763–1770, 2015.

[14] Y. Shen and H. Jin, "EpipReC: towards practical differentially private framework for personalized recommendation," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 180–191, 2016.

[15] H. Shin, S. Kim, J. Shin, and X. Xiao, "Privacy enhanced matrix factorization for recommendation with local differential privacy," IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 9, pp. 1770–1782, 2018.