2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology

Pistis: A Privacy-Preserving Content Recommender System for Online Social Communities

Dongsheng Li[†], Qin Lv[§], Huanhuan Xia[†], Li Shang[§], Tun Lu[†], Ning Gu[†]
[†] Fudan University [§] University of Colorado at Boulder

***Abstract*—With the explosive growth of online social communities and massive user-generated content, privacy-preserving recommender systems, which identify information of interest to individual users without disclosing personal interests to other parties, have become increasingly important. Collaborative filtering (CF), a widely used recommendation technique, recommends content that similar users have liked. As a result, CF-based recommender systems may expose sensitive personal interest information. This is demonstrated by a privacy attack model we present that targets online social communities.**

To solve this problem, we propose an interest group based privacy-preserving recommender system called *Pistis*. By identifying inherent item-user interest groups and separating users' private interests from their public interests, *Pistis* can make recommendations based on aggregated judgments of group members and local personalization, thus avoiding the disclosure of personal interest information. Pistis has been deployed and evaluated in an online social community with over 63,000 users, 20,000 daily posts, and 180,000 daily reads. Compared with two representative CF-based methods, our evaluation results demonstrate that Pistis achieves better performance in privacy preservation, recommendation quality, and efficiency.

***Keywords*-recommender system; privacy-preserving; online social community**

I. Introduction

During the recent years, online social communities have gained explosive popularity and are now among the most visited websites on the Internet. Everyday, large amounts of information are generated and accessed by individual users in online social communities. More and more users rely on recommender systems to identify content items that are of interest to them. Most recommender systems are designed to identify the interests of individual users [1], [2]. Such personal interest information may be sensitive and private. Privacy-preserving recommender systems have become increasingly important for online social communities.

Collaborative filtering (CF), one of the most commonly used recommendation techniques, has been an area of active research [3]. CF-based methods recommend items based on the idea that like-minded users in the past may have similar taste in the future. CF-based recommender systems typically contain two stages: The first stage collects user preference information over various content items and identifies users with similar interests; and the second stage makes recommendations based on such user interest correlations.

Serious privacy issues may arise in this process, as personal sensitive information is collected by the central server. This problem is investigated by recent works on privacy preserving collaborative filtering (PPCF) [4], [5], [6]. Moreover, new privacy issues arise in online social communities, as users expose part of their interests when they post or comment on content items. A malicious user can create pseudo online identities and try to mimic the content interests of a target user by deliberately selecting and reading the target user's public posts and comments — therefore creating a strong interest correlation with the target user. Since existing CF-based (or PPCF-based) recommender systems do not distinguish users' disclosed interests from their private ones, these systems would recommend to the malicious user items that reflect the target user's interests, including those that the user does not intend to disclose. For example, Alice reads and posts articles about traveling, which is fine for others to know. But she does not want others to know that she also reads about delinquency. Bob, a malicious user trying to identify Alice's hidden interests, can read all the articles that Alice has posted or commented on, thus establishing a high similarity with Alice's interests and being recommended delinquency-related articles that Alice may have read. This problem is further demonstrated by the formal privacy attack model we propose in Section II.

In this work, we propose *Pistis*, an *interest-group* based PPCF system which can effectively address both the original privacy issue and the privacy attack problem above. In this system, all users' information is kept on their local machines, thus avoiding personal sensitive information being collected by the central server. Moreover, items are clustered into distinct interest groups, which effectively break the ties between private and public interests of online users. Content ratings are determined via distributed secure multi-party computation, recommendations are generated based on interest groups and further personalized at each user's local machine, thus concealing sensitive interests of users yet achieving high recommendation quality. To the best of our knowledge, this is the first work identifying and addressing the personal interest privacy preservation problem of CF-based recommender systems for online social communities. Our key contributions are as follows:

1. We propose a privacy attack model to identify and quantify the personal interest privacy problem in online social communities and CF-based recommender systems.

2. We propose a novel secure multi-party summation protocol, which can compute the sum of n private values held by n parties without exposing any party's private value.

3. We propose an interest group based recommendation algorithm, in which ratings of content items are calculated within interest groups and personalization is done on the

978-0-7695-4513-4/11 $26.00 © 2011 IEEE
DOI 10.1109/WI-IAT.2011.136

client side, thus protecting user privacy from other parties, and are more robust under the privacy attack model.

4. We deploy and evaluate the proposed system in an online social community with over 63,000 users, 20,000 daily posts and 180,000 daily reads. Our experimental results demonstrate that the proposed recommendation method outperforms two state-of-the-art CF methods in privacy preservation, recommendation quality, and efficiency.

The rest of this article is organized as follows. Section II analyzes the privacy issues of CF-based methods in online social communities and proposes a privacy attack model. Section III presents in detail the proposed interest group based privacy preserving recommender system, including interest group identification, interest group based recommendation, and local personalization. System deployment and evaluation results are presented in Section IV. Section V discusses related work, and Section VI concludes.

II. Problem Formulation

In this section, we first analyze the user interest privacy issues in online social communities and CF-based recommender systems. We then propose a privacy attack model targeting CF-based recommender systems for online social communities. A quantitative analysis of privacy leakage is demonstrated using a real-world case study.

A. Privacy in Online Social Communities

In online social communities (e.g., Facebook, Twitter, etc.), the diversity and massive size of user-generated content and close inter-user content interactions have raised new concerns on recommender systems, among which user interest privacy preservation is a key challenge.

While various definitions of privacy exist in the literature and general practice, for the purpose of recommending content items in online social communities, we define two notions of user interest: $I(u)_{public}$ and $I(u)_{private}$.

Definition 1. *The public interest* $I(u)_{public} = \{i_1, i_2, \cdots, i_x\}$ *of user* u *is a set of items. Item* $i \in I(u)_{public}$ *iff.* u *has publicly posted* i *or commented on* i*.*

Definition 2. *The private interest* $I(u)_{private} = \{j_1, j_2, \cdots, j_y\}$ *of user* u *is a set of items. Item* $j \in I(u)_{private}$ *iff.* u *has read* j*,* $j \notin I(u)_{public}$*, and* $\forall i \in I(u)_{public}$*,* i *and* j *do not belong to the same interest group.*

We define $I(u)_{public}$ as the information that a user u chooses to disclose (i.e., via his/her own posts or comments on other users' posts), while $I(u)_{private}$ represents the "hidden" information that u reads but does not want others to know. In other words, a user's personal interests can be separated into two categories: *public* and *private* interests.

Our goal is to *protect user interest privacy when recommending content items to individual users in online social communities*. Specifically, we focus on protecting individual users' *private interests*, since these are the information that users do not disclose and wish to keep private. *Such user interest privacy should be protected against both the central server (service provider) and any third parties.*

B. Privacy in Collaborative Filtering

Collaborative filtering (CF) has been widely used in content recommender systems [1], [2], [7], [8], [9], thanks to its high accuracy and robustness. By examining the content item preferences (ratings) of users in the past, CF-based recommender systems can identify inherent similarities among users' content interests and make predictions about content items that a user may like. In the process of identifying shared user interests and leveraging such information for content recommendations, sensitive personal interests of specific items are revealed to the server, leading to serious user privacy implications.

Several techniques have been proposed to prevent the server from learning user-specific interests, such as secure multi-party computation and randomized perturbation [4], [10], [5], [11], [12], [6]. However, these techniques do not consider information disclosed by users in online social communities, nor do they prevent the server or third parties from inferring private user interests, as we demonstrate in the following formal privacy attack model and case study. Intuitively, by following a target user A's disclosed interests $I(A)_{public}$, a malicious user B can establish a high interest similarity with A, and a standard CF-based recommender system would recommend to B items that A has liked, including the items in $I(A)_{private}$.

C. Privacy Attack Model

Based on the analysis above, we propose an attack model that aims to learn users' private interests in online social communities by probing a CF-based recommendation server.

Consider an online social community and its associated recommender system. The following operations are allowed by any user: a) creat a user account; b) post or comment on a content item; c) read the content items posted or commented on by other users; and d) request recommendations. Let U be the set of users and I be the set of items.

Definition 3. *The* Privacy Attack Model *is a 3-tuple* $M(u, v) = \{I(u)_{public}, I(u)_{private}, R_v\}$*, where* $u \in U$ *is the target user,* $v \in U$ *is the pseudo user identity created by an attacker.* $I(u)_{public}$ *and* $I(u)_{private}$ *are defined as above, which are the public and private interest of user* u*.* $R_v \subseteq I$ *is the set of items that are recommended to* v*.*

The attacker works as follows:

- Step 1: v identifies and "reads" all items in $I(u)_{public}$;
- Step 2: v requests recommendations R_v and only "reads" the items that either rank high in the recommendation list or are participated (posted or commented on) by u;
- Step 3: Repeating Step 2 multiple times, the items in $R_v - I(u)_{public}$ gradually approximate items in $I(u)_{private}$.

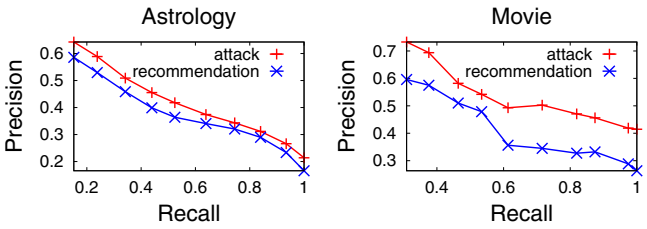

Figure 1. Case study: Privacy attack of user private interests in online social communities (two subcommunities) with CF-based recommendation.

Intuitively, using this privacy attack model, an attacker can "mimic" the target user's publicly-disclosed interests $I(u)_{public}$, and through the iterative recommendation process, become more similar to the user's overall interests, and eventually identifies the user's private interests $I(u)_{private}$.

D. Case Study

To further validate and quantitatively analyze the user private interest leakage problem of CF-based recommender systems for online social communities, we have conducted a case study on Fudan BBS (http://bbs.fudan.edu.cn), a popular online social community among Chinese universities. It has over 63,000 users and supports various content-related user interactions, including posting, reading, and replying to articles and multimedia content. Everyday, there are approximately 20,000 new posts and 180,000 reads.

Using Fudan BBS and a state-of-the-art CF-based recommendation method [2], we evaluate the effectiveness of the proposed attack model. The attack model "guesses" users' private interests by establishing high similarity with the target users, in order to "trick" the recommender system to use the target user's interests (both public and private) to make recommendations for the attacker. For each target user u, we consider the items in $I(u)_{private}$ as the target items. The attacker aims to "guess" a set of items $I(u)_{attack}$ as close to $I(u)_{private}$ as possible. To evaluate the quality of the attack, we define the following two metrics: $AttackPrecison = \frac{|I(u)_{attack} \cap I(u)_{private}|}{|I(u)_{attack}|}$ and $AttackRecall = \frac{|I(u)_{attack} \cap I(u)_{private}|}{|I(u)_{private}|}$. $AttackPrecision$ refers to the fraction of items identified by the attacker that are true hidden items of the user, and $AttackRecall$ refers to the fraction of hidden items that are exposed by the recommendation server and identified by the attacker. Higher $AttackPrecision$ and/or higher $AttackRecall$ thus indicate more effective attack and more severe privacy leakage.

As shown in Figure 1, the privacy attack model we propose can effectively identify the hidden items, achieving high attack precision and recall. For comparison purposes, the effectiveness of the attack model is further compared with the CF-based recommendation method itself. The CF-based recommendation server has the complete knowledge of the target users' past activities and interests, and "guesses" the users' interests based on the activities of other online users with similar interests. It is intriguing to see that, the attack model achieves a higher precision and recall than the overall recommendation quality of the CF-based recommendation server. This is due to the fact that the attack model can construct very high correlations with the target users, and the recommendation results for the attacker are more biased towards the target users' interests. Privacy leakage is thus a serious challenge that needs to be addressed.

III. The Pistis Recommender System

We propose *Pistis*, an interest group based privacy-preserving recommender system, to protect users' private interests in online social communities against the privacy attack model discussed above. As shown in Figure 2, the Pistis system consists of two key components:

• *Privacy-preserving interest group identification (step (1) in Figure 2).* This component clusters items and users into interest groups, such that each interest group contains items of similar content and users who share that interest. A user can belong to multiple interest groups, thus allowing the separation of a user's public interests from his/her private interests. Our novel secure multi-party computation protocol ensures user privacy when obtaining interest groups.

• *Privacy-preserving interest group based recommendation (steps (2) and (3) in Figure 2).* Using the interest groups we have identified, the second component determines each item's rating within an interest group through aggregated and privately weighted voting of individual users in that interest group. The item ratings represent a group of users' interests and protect individual users' private interests. The item ratings are further personalized on each client to generate final recommendations for its user, thus achieving both high recommendation quality and good privacy preservation.

A. Privacy-Preserving Interest Group Identification

To protect user privacy, the key is to break the correlation between users' exposed interests and private interests. To manage the various types of user interests, we propose the notion of *interest groups*, as defined below:

Definition 4. *The set of interest groups* $G = \{g_1, g_2, ..., g_k\}$*, where* k *is the number of interest group in* G*, has the following properties: (1) Each* $g \in G$ *is a 3-tuple* $g =< I_g, U_g, c_g >$*, in which* $I_g = \{i_1, i_2, ..., i_m\}$ *is a set of items,* $U_g = \{u_1, u_2, ..., u_n\}$ *is a corresponding set of users, and* $c_g \in I_g$ *is the center of* g*. (2) For each user* $u \in U_g$*,* u *likes the items in* I_g*. (3) Center* c_g *has the smallest*

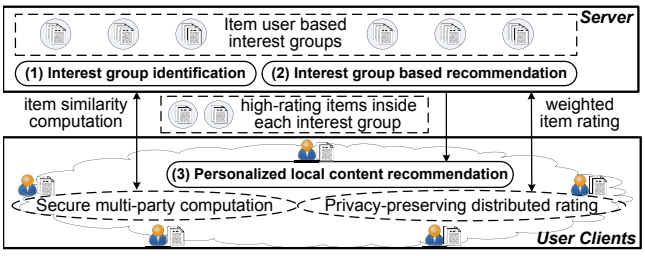

Figure 2. Pistis: An interest group based privacy-preserving content recommender system for online social communities.

average distance from other items in I_g, and it represents the "interest" of g. (4) For any two interest groups g_i and g_j ($1 \leq i, j \leq k$ and $i \neq j$), $I_{g_i} \cap I_{g_j} = \emptyset$ and $U_{g_i} \neq U_{g_j}$.

There are two main challenges in identifying interest groups: 1) representativeness of interest groups, i.e., good intra-group similarity and inter-group separation; and 2) protection of user interest privacy in the process of interest group identification. To address these issues, we adopt the k-centroids clustering algorithm [13] as follows:

1. Randomly select k items as the k centroids.

2. Calculate the distance between the k centroids and each item. Assign each item to the cluster with the closest centroid. Inside each cluster, choose the item with the smallest average distance to other items as the new centroid.

3. Repeat step 2 until the k centroids do not change.

In k-centroids clustering, only the distance calculations among items are required. If we can calculate the distances with privacy protection, we can identify the interest groups without disclosing user interests.

1) Privacy-Preserving Distance Calculation: Given a set of users U and a set of items I, our goal is to cluster the items (and the corresponding users) into k interest groups. We first consider the distance/similarity function between two items. This function should adequately capture the similarity of users' interests in different items, and should be easy to calculate in a privacy-preserving and distributed fashion. Specifically, we leverage the Jaccard similarity. Let U_i (U_j) be the set of users who are interested in item i (j), then $ItemSimilarity(i, j) = |U_i \cap U_j|/|U_i \cup U_j|$. Using this similarity function, items who are of interest to the same set of users will have high similarity and be clustered into the same group, i.e., the interest group.

To protect user privacy, a user's interests in specific items are stored locally on each user's machine and are not disclosed to other parties including the server machine. Therefore, a secure multi-party computation mechanism is needed to compute the similarity between any two items. We present in Section III-A2 the $SecureSum()$ function, which can compute the sum of n parties' private values without disclosing the private values. To utilize this function, we need to convert the set operations of $|U_i \cap U_j|$ and $|U_i \cup U_j|$ into n-party summations. This is achieved by defining each user's interest on a specific item as 1 (interested) or 0 (not interested). The detailed computation is shown in Algorithm 1.

Given the $ItemSimilarity()$ function, for a given k, the server can then adopt the k-centroids clustering algorithm to cluster the items into k different interest groups. Once items are separated into k interest groups, a user's interests are then determined by the number of items he/she likes in each interest group, and the user's private interests correspond to the interest groups in which no item has been publicly disclosed by the user via posting or commenting.

2) Secure Multi-Party Computation: The algorithms above utilize the $SecureSum()$ protocol, which can calculate the sum of n private values held by n parties without exposing any of the values. This protocol is derived from secure multi-party computation (SMPC), which was studied first by Yao in his famous Yao's millionaire problem [14] and later by Goldreich [15]. Recently, Canny proposed a secure multi-party computation method on encrypted data to achieve private collaborative filtering [4]. His method adopted homomorphic encryption and distributed threshold decryption. However, the decryption phase requires that more than a predefined fraction of users must be online together in order to decrypt the data. A higher predefined fraction threshold improves system security but is difficult to reach in practical use. In Pistis, we design a new secure multi-party summation protocol which can compute the summation of n values held by n parties without exposing the input of any party, and this protocol does not require data encryption and decryption. The details of the $SecureSum()$ protocol are described in Algorithm 2.

Algorithm 1 $ItemSimilarity(i, j, U)$

Require: $i, j \in I, i \neq j$, for any user $u \in U$, $r_{u,i}$ is u's rating of item i, $r_{u,i} = 1$ if u is interested in item i, and 0 otherwise

1: $cap = 0, cup = 0$
2: use $SecureSum()$ to compute $cap = \sum_{u \in U}(r_{u,i} \cdot r_{u,j})$
3: use $SecureSum()$ to compute $cup = \sum_{u \in U}(r_{u,i} \oplus r_{u,j})$
4: return $similarity = cap/cup$

Algorithm 2 $SecureSum(P, M, r)$

Require: $|P| = |M| = n$ $(n > 1)$, $i \in [1, n]$, $p_i \in P$ is the i-th party, $m_i \in M$ is the private data of the i-th party. r is a predefined number of rounds in the protocol.

1: each party p_i selects a random number $r_i \geq r$, then divides its private data m_i into r_i random parts such that the sum of the r_i parts is m_i. All the r_i parts form a local data set D_i
2: **while** $r > 0$ **do**
3: each party p_i randomly selects a data $d_i \in D_i$ and a party $p_j \in P$, then p_i sends d_i to p_j
4: each party p_i puts all the received data into D_i
5: $r = r - 1$
6: **end while**
7: each party p_i sums all the remaining local data in D_i to get the local sum sum_i, then sends sum_i to the server
8: server returns $sum = \sum_{1 \leq i \leq n} sum_i$

B. Privacy-Preserving Interest Based Recommendation

Given the interest groups we have identified above, we can then make recommendations at run-time for individual users, while protecting their private interest information. To achieve this, our recommendation algorithm works in three stages: 1) identify the interest group that an item belongs to; 2) gather aggregated and weighted rating for each item from all users within the corresponding interest group; and 3) on the client machines, the interest group based item ratings are personalized and ranked to generate recommendations for their specific users. By assigning items into interest groups and generating group-based item ratings, we can effectively

separate users' public interests from private interests and protect users' private interests. The final personalization step also ensures good recommendation quality.

Note that the interest group that an item belongs to can be determined using the privacy-preserving $ItemSimilarity$ algorithm in the previous section. In this section, we focus on the calculation of weighted item ratings within interest groups and local personalized item ranking with privacy.

1) Privacy-Preserving Distributed Item Rating: First, we define the weight (importance) of each user u in a given interest group g that item i belongs to. Let I_g be the set of items that belong to interest group g, and I_u be the set of items that user u is interested in. We define

$$Precision(u,g) = \frac{|I_u \cap I_g|}{|I_g|},\ Recall(u,g) = \frac{|I_u \cap I_g|}{|I_u|} \quad (1)$$

$$F{-}measure(u,g) = \frac{2 \cdot Precision(u,g) \cdot Recall(u,g)}{Precision(u,g) + Recall(u,g)} \quad (2)$$

Here, $Precision(u,g)$ measures the fraction of items in g that u likes, and $Recall(u,g)$ measures the fraction of items liked by u that are actually in g. Both $Precision(u,g)$ and $Recall(u,g)$ are necessary to measure a user's importance inside a group, so we adopt their combination, $F - meausre(u,c)$, as the weighted importance measure of user u in interest group g, i.e., $Weight(u,g) = F - measure(u,g)$. Please note that, $Weight(u,g)$ can be computed and stored by each user locally.

To compute the weighted rating of item i in group g, we need to aggregate individual users' ratings of i in a distributed fashion, with privacy preservation. This is achieved via the $SecureSum()$ protocol presented earlier. The item rating within interest group is determined as follows:

$$ItemRating(i) = \frac{\sum_{u \in U} Weight(u,g) \times r_{u,i}}{\sum_{u \in U} Weight(u,g)} \quad (3)$$

Here, $r_{u,i}$ is user u's rating of item i, $r_{u,i} = 1$ if u is interested in item i, and 0 otherwise. The nominator and denominator of Equation 3 can be obtained separately by $SecureSum$ protocol, then the server can obtain the rating of i without gathering the private interests of each user.

2) Local Recommendation Generation: After receiving the aggregated item ratings from the server, a final step of local recommendation generation needs to be performed on each user's machine. This step combines the interest group based item ratings with a user's specific interests (maintained securely on his/her local machine) to generate a list of items that are most likely to be of interest to the specific user. This step is necessary to ensure high recommendation quality.

Although it is relatively easy to rank items that belong to the same interest group, for items that belong to different interest groups, their aggregated item ratings are not directly comparable, as interest groups vary significantly in size and the user may have different levels of interest in different groups. To address this issue, we adopt a weighted local ranking mechanism, which weighs the ratings of different items by the user's interest level in the groups that contain the items. The details are shown in Algorithm 3 .

Algorithm 3 $LocalRanking(u, G, I', A)$

Require: u is a user G is the set of interest groups, I' is the set of items to be recommended, A is the set of aggregated ratings of all items in I'

1: **for** each $g \in G$ **do**
2: u locally computes the fraction of items in g that are liked by u, denoted as $w_{u,g}$.
3: **end for**
4: **for** each $i \in I'$ **do**
5: find the interest group g that contains i
6: $score_{u,i} = w_{u,g} \times A_i$
7: **end for**
8: recommend items with the top $score_{u,i}$ values to u

C. Discussion

1) Complexity Analysis: To generate recommendations for n users and m items, traditional centralized CF-based recommender systems have a computation complexity of at least $O(mn)$, and some of them are more complex than $O(mn^2)$. The Pistis recommender system has a computation complexity of $O(mn)$, which is the same as the most efficient CF methods in existence. The communication complexity of Pistis is slightly higher than that of the centralized CF-based methods, because we need to determine the interest group that a target item belongs to before making recommendations. Specifically, the communication complexity of centralized CF methods is $O(mn)$, because they send m items to n users. The communication complexity of our method is $O(m(n+k))$ (k is the number of interest groups), as the communication complexity of determining m items' memberships in k interest groups is $O(mk)$. Since k is usually much smaller than n, the extra communication overhead of Pistis is negligible.

2) User Interest Privacy: In this work, we adopt the semi-honest model, in which each participant records all temporary values of the computation, but does not dramatically change the input in different rounds. This model is suitable for our scenario as a malicious user can not obtain further benefits, i.e., more information of the targeted user's private interest, by manipulating his local data.

In this model, no one can get all the inputs of another user in the $SecureSum$ protocol, which means all users' privacy is preserved during the computation process. In the semi-honest model, the privacy-preservation feature of $SecureSum$ protocol can be proved formally, but due to the page limit, we omitted the proof. Thus, the interest group identification and recommendation algorithm presented in the previous sections, which are based on the $SecureSum$ protocol, are privacy preserving.

IV. Evaluations

This section evaluates Pistis, the proposed interest-group based privacy-preserving recommender system for online

social communities. Pistis has been implemented and deployed in Fudan BBS, which is introduced in Section II-D. Pistis is running on a Dell blade server equipped with an Intel quad-core 2.4 GHz CPU and 32 GB memory. In addition, a user client software is implemented as a user-friendly browser plug-in, interacting with its user (owner), recommendation server, and other clients. Our evaluations consider eight of the most popular subcommunities in Fudan BBS (see Figure 3). For each study, three-week usage data is considered, with the first two-week data for training and the third-week data for testing. Pistis is compared against three CF algorithms. One is the MinHash-based CF algorithm (MCF), which is a scalable CF algorithm and achieves comparable recommendation quality with a PLSI-based CF algorithm [2]. The other is an SVD-based CF algorithm (SVD), which is a privacy-preserving CF algorithm proposed by Canny [4]. This method adopts homomorphic encryption and distributed threshold decryption to achieve secure multi-party computation, which requires lots of communication and computation. And the third one is a classic CF algorithm [16], but the user similarities and predictions are calculated using only users' public interests (i.e., best protection of user interest privacy). We refer to this method as the Basic CF method (BCF). The comparisons primarily focus on user interest privacy protection, content recommendation quality, and system scalability. Our comparative evaluations draw the following conclusions.

• *Recommendation quality:* Pistis can significantly improve the content recommendation quality – on average, 56% over MCF, 47% over SVD, and 164% over BCF.

• *Privacy preservation:* Pistis can effectively improve user private interest protection, with an average of 157% improvement over MCF and 139% improvement over SVD.

• *Efficiency:* Pistis is distributed. With the coordination of the server, content recommendation related computation is offloaded to clients, which offers much better system scalability. Compared against MCF, Pistis reduces the server computation workload by 91% on average, with only 6% increase of communication overhead. Compared against SVD, a pure peer-to-peer model, Pistis reduces the computation workload by 26% and communication workload by 57%.

A. Recommendation Quality

Using the selected eight subcommunities, the recommendation quality of Pistis is evaluated and compared against that of MCF, SVD and BCF. The results are shown in Figure 3. As we can see, BCF, which has the best user privacy preservation, performs much worse than the other three algorithms. This is due to the limited amount of information that BCF uses, i.e., users' public information alone will not be sufficient to make accurate recommendations. So we will not consider BCF in later comparisons.

Pistis outperforms MCF in all 8 subcommunities, and the corresponding improvement is between 5% and 213% (56% on average). Pistis outperforms SVD in all 8 subcommunities, and the corresponding improvement is between 5% and 191% (47% on average). Using Pistis, better recommendation quality can be achieved because (1) the proposed interest group based recommendation solution enables highly selective content recommendation only to the interested users, i.e., users within the corresponding interest groups, and (2) the quality of each item is more accurately determined only by the interested users, and the noisy inputs from other irrelevant users are avoided.

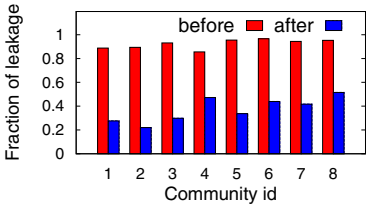

Figure 4. User interest leakage before and after user interest group clustering. The x label is defined as following, 1: Astrology, 2: Auto, 3: Music, 4: Movie, 5: Joke, 6: OMTV, 7: TV, 8: TVEntZ, and similarly hereinafter.

B. Privacy Preservation

The first experiment evaluates user exposed interest before and after interest group clustering. As shown in Figure 4, before user interest group clustering, most of the user interests (92.2% on average) are exposed as the hidden user interests are correlated with their public interests. However, after user interest group clustering, only a small fraction of user hidden interests (40.1% on average) are exposed to the attackers, which means a bigger fraction (52.1% on average) of user hidden interests are protected by the clustering and separation of interest groups.

The second experiment evaluates user interest privacy preservation of different algorithms. For each of the eight subcommunities, we consider users who have had post activities (i.e., disclosed public interests) in that subcommunity. For each target user A, a malicious attacker B tries to obtain A's private interests by following the attack procedure of the proposed privacy attack model – B reads A's public posts during the first two weeks, and then determines A's private interests based on the system recommendation results during the third week. The resulting A's private interest breach is quantified by $AttackPrecision$ and $AttackRecall$ as defined in Section II.

For each online subcommunity, the average attack measures are then calculated, and the results are shown in Figure 5. We see that Pistis can effectively improve the online users' interest privacy, reduce the attack precision by 157% on average (55% minimum and 362% maximum) over MCF, and reduce the attack precision by 139% on average (47% minimum and 293% maximum) over SVD. Note that, the above defined privacy measures are pessimistic for Pistis, as some (or even many) of the items read by A and disclosed by Pistis may belong to the same interest groups as some of A's public posts. These disclosed read items reflect A's public interests rather than A's private interests. Therefore, such information disclosure has limited impact on

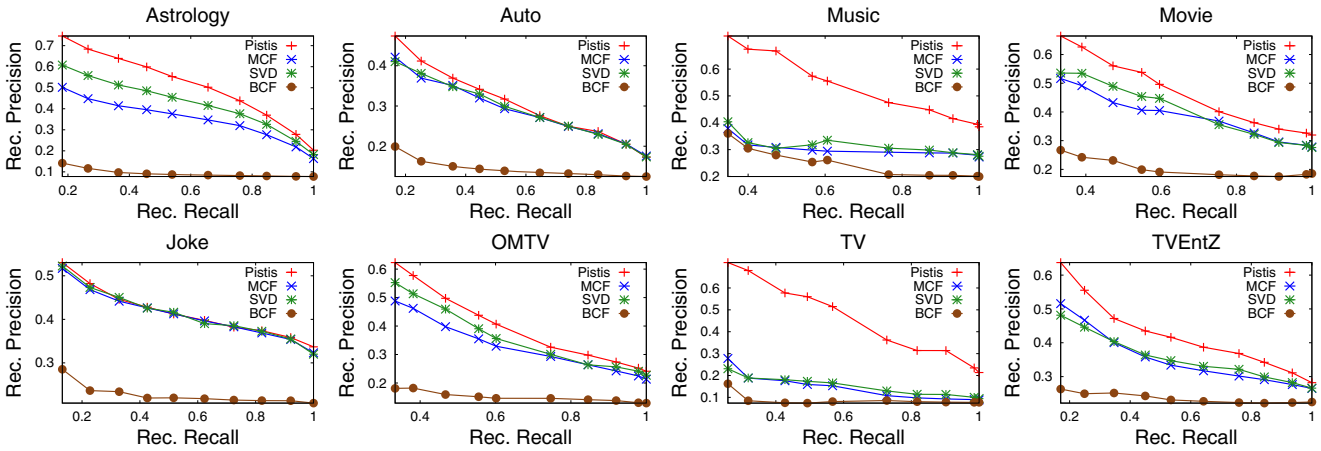

Figure 3. Recommendation quality of Pistis, MCF, SVD and BCF in eight subcommunities.

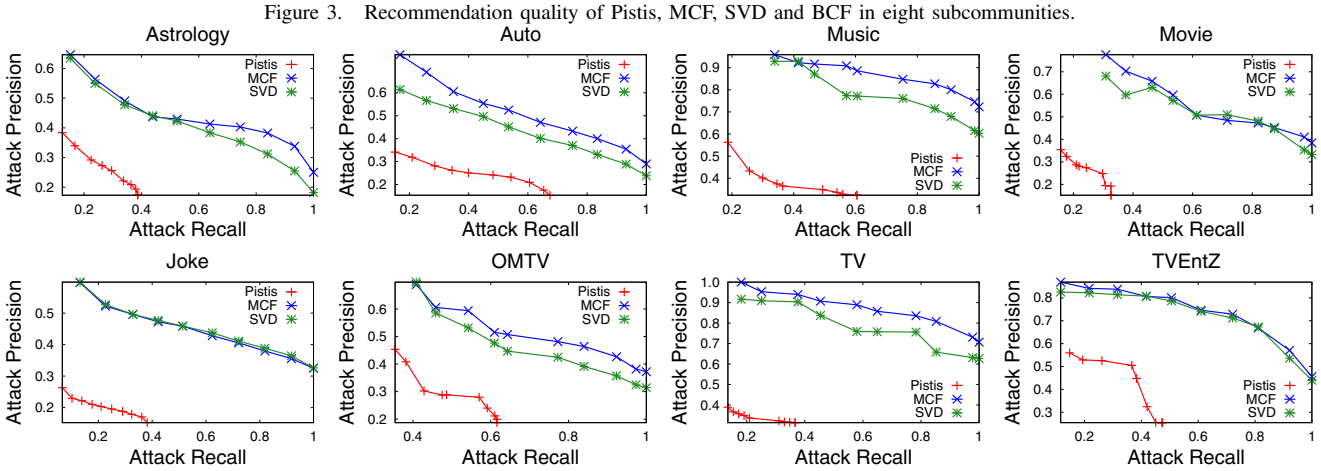

Figure 5. Privacy preservation analysis of Pistis, MCF and SVD in eight subcommunities.

A's private interest breach. On the other hand, the interest groups, which reflect A's private interests rarely contain A's public posts, will never be disclosed to B by Pistis.

C. Recommendation Efficiency

In the efficiency analysis, we first compare Pistis with SVD, a distributed privacy-preserving CF algorithm, which is similar to Pistis. We compare the overall computation and communication overhead of the two algorithms in Figure 6. As we can see, Pistis requires 26% less computation overhead and 57% less communication overhead compared with SVD. The per user computation and communication overhead comparison is similar to the above results. This demonstrates that Pistis is more efficient than SVD.

Next, we compare the efficiency of Pistis to that of MCF, which is a centralized CF algorithm without privacy preservation. To achieve privacy preservation, Pistis adopts the $SecureSum()$ protocol to ensure secure multi-party summation, which increases the communication overhead. Figure 7 (a) shows the overall efficiency comparison. As we can see, the computation overhead of Pistis and MCF are comparable. However, Pistis requires more communication than MCF due to the distributed design and communication needs of the $SecureSum()$ protocol. This is acceptable as the communication overhead is shared among the server and a large number of user clients, while the benefits in privacy preservation and recommendation quality are more significant. Since the server is often the bottleneck in many recommender systems, we further compare the sever-side computation and communication overhead of Pistis and MCF in Figure 7 (b). Compared with MCF, Pistis requires much less computation, 91% less on average. Also, the server-side communication of Pistis and MCF are comparable, and Pistis requires only 6% more communication on average compared with MCF. The high efficiency of Pistis on the server side is achieved because the computation and communication are mostly done on the distributed clients. This demonstrates that Pistis is much more scalable than MCF due to its lower server-side overhead. This is particularly important for online social communities, since the recommendation server has to handle a large number of items and support a large number of users.

V. Related Work

Our work of privacy-preserving content recommendation for online social communities builds upon previous research in several related areas, including privacy-preserving collaborative filtering, secure multi-party computation and other

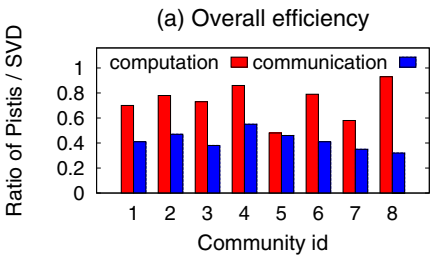

Figure 6. Efficiency comparison of Pistis and SVD.

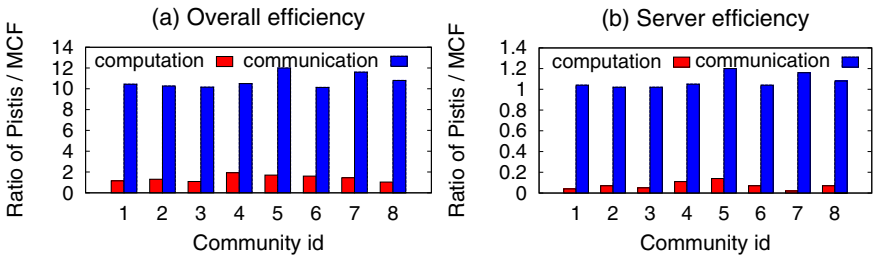

Figure 7. Efficiency comparison of Pistis and MCF in eight subcommunities.

issues in recommender systems.

Existing works on privacy-preserving collaborative filtering (PPCF) use secure multi-party computation [4], [10] or randomization / obfuscation techniques [5], [11], [12], [17], [6], [18] to avoid disclosing personal information to the central server. None of them considers user interest privacy protection in online social communities, where users choose to disclose some of their interests via posting and commenting. While these PPCF methods can preserve user privacy from being gathered by central servers, they cannot prevent malicious users from obtaining users' private interests by exploiting inter-user interest correlations, an essential basis of CF-based recommender systems. Ahn et al. [19] proposed a distributed and privacy-preserving expert CF method, in which users download expert ratings and generate recommendations locally. But expert information, which is difficult to obtain in online social communities, is not required in our work.

In this work, we propose a secure multi-party summation protocol, which shares similar idea with the k secure sum protocol proposed by Sheikh et al. [20]. We also adopt the idea of breaking private data into segments. However, our protocol uses different data obfuscation techniques and requires much less communication overhead, and is thus more efficient in large-scale distributed computation.

VI. Conclusions

The massive amounts of user-generated content and the close content interactions among users in online social communities raise unique and serious privacy concerns. CF-based recommender systems, leveraging user interest correlations for content delivery, further exacerbate the privacy problem. This article tackles the privacy-preservation problem of CF-based recommender systems for online social communities. We propose a privacy attack model to identify and quantify the leakage of the private interests of online users caused by existing CF solutions. To protect user interest privacy, we propose *Pistis*, an interest-group based content recommender system for online social communities. Real-world deployment and detailed evaluation results demonstrate that, compared with state-of-the-art CF solutions, our solution offers better privacy preservation, recommendation quality, and efficiency.

Acknowledgment This work is supported by the National Natural Science Foundation of China under Grant No. 60736020 and No. 60803118, the Shanghai Leading Academic Discipline Project under Grant No. B114, and the National Science Foundation under Grant No. CNS–0910995.

References

[1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Trans. on Knowl. and Data Eng.*, vol. 17, no. 6, pp. 734–749, 2005.

[2] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: scalable online collaborative filtering," in *WWW '07*, 2007, pp. 271–280.

[3] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Advances in Artificial Intelligence*, vol. 2009, article ID 421425, 19 pages, 2009.

[4] J. Canny, "Collaborative filtering with privacy," in *S&P '02*, pp. 45–57.

[5] H. Polat and W. Du, "Privacy-preserving collaborative filtering using randomized perturbation techniques," in *ICDM '03*, pp. 625–628.

[6] F. McSherry and I. Mironov, "Differentially private recommender systems: building privacy into the net," in *KDD '09*, 2009, pp. 627–635.

[7] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, pp. 61–70, 1992.

[8] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "Grouplens: an open architecture for collaborative filtering of netnews," in *CSCW '94*. ACM, 1994, pp. 175–186.

[9] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: Item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.

[10] J. Canny, "Collaborative filtering with privacy via factor analysis," in *SIGIR '02*. ACM, 2002, pp. 238–245.

[11] S. Zhang, J. Ford, and F. Makedon, "A privacy-preserving collaborative filtering scheme with two-way communication," in *EC '06*. ACM, 2006, pp. 316–323.

[12] S. Berkovsky, Y. Eytani, T. Kuflik, and F. Ricci, "Enhancing privacy and preserving accuracy of a distributed collaborative filtering," in *RecSys '07*, 2007, pp. 9–16.

[13] F. Leisch, "A toolbox for k-centroids cluster analysis," *Comput. Stat. Data Anal.*, vol. 51, pp. 526–544, 2006.

[14] A. C. Yao, "Protocols for secure computations," in *FOCS' 82*. IEEE Computer Society, 1982, pp. 160–164.

[15] O. Goldreich, "Secure multi-party computation," Final(incomplete) Draft, Version 1.4, Oct 2002.

[16] J. L. Herlocker, J. A. Konstan, A. Borchers, and J. Riedl, "An algorithmic framework for performing collaborative filtering," in *SIGIR '99*, 1999, pp. 230–237.

[17] S. Zhang, J. Ford, and F. Makedon, "Deriving private information from randomly perturbed ratings," in *SDM' 06*, 2006, pp. 59–69.

[18] R. Shokri, P. Pedarsani, G. Theodorakopoulos, and J.-P. Hubaux, "Preserving privacy in collaborative filtering through distributed aggregation of offline profiles," in *RecSys '09*. ACM, 2009, pp. 157–164.

[19] J.-w. Ahn and X. Amatriain, "Towards fully distributed and privacy-preserving recommendations via expert collaborative filtering and restful linked data," in *WI-IAT '10*, pp. 66–73.

[20] R. Sheikh, B. Kumar, and D. K. Mishra, "A distributed k-secure sum protocol for secure multi-party computations," *Computing Research Repository (CoRR)*, pp. 68–72, 2010.