Enhanced Middleware for Collaborative Privacy in Community Based Recommendations Services

Ahmed M. Elmisery, Kevin Doolin, Ioanna Roussaki and Dmitri Botvich

Abstract Recommending communities in social networks is the problem of detecting, for each member, its membership to one of more communities of other members, where members in each community share some relevant features which guaranteeing that the community as a whole satisfies some desired properties of similarity. As a result, forming these communities requires the availability of personal data from different participants. This is a requirement not only for these services but also the landscape of the Web 2.0 itself with all its versatile services heavily relies on the disclosure of private user information. As the more service providers collect personal data about their customers, the growing privacy threats pose for their patrons. Addressing end-user concerns privacy-enhancing techniques (PETs) have emerged to enable them to improve the control over their personal data. In this paper, we introduce a collaborative privacy middleware (EMCP) that runs in attendees' mobile phones and allows exchanging of their information in order to facilities recommending and creating communities without disclosing their preferences to other parties. We also provide a scenario for community based recommender service for conferences and experimentation results.

Keywords Privacy · Clustering · Community Recommendations · Middleware

A. M. Elmisery (✉) · K. Doolin · D. Botvich
TSSG, Waterford Institute of Technology-WIT-Co, Waterford, Ireland
e-mail: ahmedkhmais2001@yahoo.com

I. Roussaki
National Technical University of Athens, Athens, Greece

S.-S. Yeo et al. (eds.), *Computer Science and its Applications*,
Lecture Notes in Electrical Engineering 203, DOI: 10.1007/978-94-007-5699-1_32,

1 Introduction

With the popularity of social networks in the last few years, users are incited to build profiles containing their preferences, join different groups and utilize various services provided within the social platform. Community based recommender service (CRS) is a service running on social media platform and aims at providing end-users referrals to join certain sub-communities out of large number of communities that are relevant for a given end-user's interests. This service is based on the assumption that end-users with similar preferences have the same interests. CRS generates referrals based on end-user profiles containing, for each one, personal data and interests. The CRS is usually accessible and open to all attendees. However, this flexibility brings forward new threats and problems such as malicious behaviors against different participants from both service provider and other participants. For instance, malicious users may get one another's private information, such as current and previous occupations, age and relationship status, even if for the user the information is not supposed to be exposed publicly.

Several strategies have been proposed to control the disclosure of private information. The most popular approach is to permit users to maintain a set of privacy rules, according to which a decision is performed whether to release or not certain preferences in owner profile. However, these approaches are either rather coarse-grained, or require a deep understanding of the privacy control system, any change of one privacy setting may result in unwanted or unexpected behaviors. Moreover, these approaches are based on the logic of either to allow or deny releasing certain preferences in users' profiles. Once, the data is released the user have no control over it and users will be vulnerable for the privacy breaches since released pieces of users' information is often interleaved, adversaries may be able to infer other private information using inference techniques. For example work in [1] shows that private information can be inferred via social relations, and the stronger the relationships people have in the network, the higher inference accuracy can be achieved.

In this paper, we lay out recommending and creating communities functions within user-side, this privacy architecture will help foster the usage and acceptance of our proposed protocols and eliminates the risk of possible privacy abuses as the sensitive data is only available to the owner but not to any other parties. However, as a consequence of applying our protocols, the structure in data is destroyed. In order to facilitate processing of such data, our protocols maintain some properties in this data which is suitable for the required computation. In rest of this work, we will generically refer to attendees' preferences as interests. This paper is organized as follows. In Sect. 2, related works are described. Section 3 presents the proposed middleware enhanced middleware for collaborative privacy (EMCP*)* used in this work. Section 4 introduces some definition required for this paper. The proposed protocols that are used in EMCP are introduced in details in Sect. 5. In Sect. 6, the Results from some experiments on the proposed mechanisms are reported. Finally, the conclusions and recommendations for future work are given in Sect. 7.

2 Related Works

The majority of the literature addresses the problem of privacy on social recommender services, due to it being a potential source of leakage of private information shared by the users as shown in [2]. In [3] a theoretical framework is proposed to preserve the privacy of customers and the commercial interests of merchants. Their system is a hybrid recommender system that uses secure two party protocols and public key infrastructure to achieve the desired goals. In [4, 5] a privacy preserving approach is proposed based on peer to peer techniques using users' communities, where the community will have a aggregate user profile representing the group as a whole but not individual users. Personal information is encrypted and communication done between individual users but not servers. Thus, the recommendations are generated on the client side. Storing users' profiles on their own side and running the recommender system in a distributed manner without relying on any server is another approach proposed in [6].

3 The Proposed Middleware

In the scope of this work, we aim to achieve privacy by empowering an individual or group to seclude themselves or information about themselves thereby reveal themselves selectively or based on levels. We seek to achieve privacy by implementing a privacy by design approach [7] where we consider a middleware that governs data collection and processing during community building process such that attendees don't have to reveal private interests in their profiles. This will help them to control what they share with various communities and to join specific sub-community with a customized profile that access only to a subset of their interests. The intuition behind our solution stems from the fact that safest way to protect sensitive profiles data is to not publish them online, but keep them at user side. However, in order to gain most of PCRS‘s functionalities, attendees disclose their private data in some way to enable PCRS‘s functionalities.

EMCP is implemented as a middleware running on top of attendees' mobile phones [8–13]. EMCP consists of different agents each of which has a certain task, but their co-operation is required to attain the whole functionality. The local obfuscation agent creates a public profile that is used as an input to encryption agent. The encryption agent is responsible for executing two cryptographic protocols; first one is private community formation (PCF) protocol which builds general communities based on attendees' profiles, while the other one is private sub-community discovery (PSD) protocol that help to discover sub-communities inside each community. These protocols act as wrappers that conceal interests before they are shared with any external entity. EMCP requires attendees to be organized into virtual topology which may be a simple ring topology or hierarchical topology, this ordering enables them to participate in multi-party computations as well. However, PCRS (private community based recommender service) is the server that initiates the process to extract different communities and sub-communities. The scenario we are considering here is the one introduced in [8] it can be summarized as following based on conference various themes, research strategies and specific topics, the organizers setup a list of available communities on PCRS which act as interaction space that supports any interactions between attendees. Each attendee configures his EMCP to build a public profile that discloses some information about their general interests that are related to conference topics for the purpose of networking and collaboration. Attendees seek to hide from the public their specific expertise, previous conference engagements, details of their research domains and problems in hand, current and previous funded projects, sessions and presentations they are planning to attend and finally their arrival/departure times. Other Private information such as names, company, etc, by default is protected by the privacy protection laws. If attendees already belonging to previously created group, they can form a sub-community inside the conference community such that they can participate in discussions and have access to the already exchanged opinions. EMCP provides referrals to suitable sub-communities and sessions for attendees based on their interests.

3.1 Threat Model

The proposed solution is secure in an honest-but-curious model. Where, every party is obliged to follow the protocol but they are curious to find out as much as possible about the other inputs. The adversaries we consider here are untrusted CRS and malicious attendees that aim to collect other attendees' interests in order to identify and track them. Moreover we do not assume CRS to be completely malicious. This is a realistic assumption because CRS needs to accomplish some business goals and increase its revenues. Intuitively, the system privacy is high if CRS is not able to reconstruct the real attendees' private interests.

4 Problem Formulation

In the following section we outline important notions used in our previous solution in [8] and required in this work, attendees' profiles can be represented in two categories public profiles and private profile. Public profiles is a set of hypernym terms in the same semantic categories for the interests in attendee's profile [8], it represent general information that attendee configures his/her EMCP to disclose, while private profile represents the "hidden" interests that attendee does not want to disclose publically to others. Our goal is to protect private participants' profiles when formulating communities and recommending sub-communities since these are the information that attendees wish to keep private against both PCRS and third parties. The notion of community in this work can be defined:

Definition 1 A community is the set $C = \{c_1, c_2, \ldots, c_n\}$, where n is the number of sub-communities in C, has the following properties: (1) Each $\forall_{i=1}^{n} c_i \in C$ is a 3-tuple $c = \{I_c, V_c, d_c\}$ such that $I_c = \{i_1, i_2, \ldots, i_l\}$ is a set of generalized interests, $V_c = \{v_1, v_2, \ldots, v_k\}$ is a corresponding set of attendees, and $d_c \in I_c$ is the main-interest of c. (2) For each attendee $\forall_{i=1}^{l} v_i \in V_c$, v have the interests V_c. (3) d_c is the frequent interest in V_c profiles, and it represents the "core-point" of sub-community c. (4) For any two sub-communities c_a and $c_b(1 \le a, b \le n$ and $a \neq b)$, $V_{c_a} \cap V_{c_b} = \emptyset$ and $I_{c_a} \neq I_{c_b}$.

5 Proposed Privacy Enhanced Protocols for EMCP

In our architecture, privacy is attained using EMCP middleware which is hosted in attendees' mobile phones and equipped with two cryptography protocols which are private community formation protocol (PCF) and private sub-community discovery protocol (PSD) that build communities and sub-communities. EMCP allows the formation of attendees' communities; such that attendees share the same experience can engage in discussions and exchange experiences. An important requirement for our solution is the ability of an attendee to search for and join various sub-communities in private way.

5.1 Private Community Formation Protocol

Our aim is to cluster attendees' profiles into different communities. There are two challenges in identifying these communities: first one is representation of community, i.e., good intra-community similarity and inter-community separation. And the second one is the protection of private profiles in the process of community identification. In order to do so, attendees build public profiles using global information supplied by PCRS (e.g. concept taxonomy and term vocabulary) independently of their profile content, then local obfuscation agent at attendees side start mapping their profiles into this global information space to get public profiles as proposed in [8].

After building public profiles, EMCP invokes the encryption agent to execute PCF protocol that is responsible for clustering attendees into general communities, such that each general community contains various attendees who share similar interests in their profiles. An attendee can belong to multiple communities, thus allowing the separation between public profiles from his/her private profiles. Our novel secure multi-party computation protocol ensures participants privacy when forming communities and matching participant public profile with the list of available communities. PCF is executed in distrusted manor; it first creates a bag of interests representations of each attendee using their profiles data. Then, the extracted interests (words) are stemmed and filtered using domain-specific dictionary; these interests associated with a user V_c are used to create a word vector $V_c = (e_c(w_1), \ldots..e_c(w_m))$, where m is the total number of distinct words in his/her is profile, and $e_c(w_1)$ describes the degree of importance of user V_c in interest w_1 (weighted frequency). The further computation proceeds to calculate term frequency inverse profile frequency [14] as following:

$$Term-frequency_{V_c}(w_i) = \#w_i\ in\ V_c\ profile/\#words\ in\ V_c\ profile$$

$$inverse-profile-frequency_{V_c}(w_i) = log(\#user/\#profiles\ contain\ word\ w_i)$$

$$e_c(w_1) = Term-frequency_{V_c}(w_i) * inverse-profile-frequency_{V_c}(w_i)$$

The similarity function between two attendees' profiles data should adequately capture the similarity of attendees' interests, and should be easy to calculate in a distributed and private fashion. Specifically, we leverage the Dice similarity for this task. Let $\mathrm{V_c}(\mathrm{V_d})$ be the two word vectors for attendees C and D then:

$$UsersSimilarity(V_c, V_d) = 2|V_c \cap V_d|/|V_c|^2+|V_d|^2$$

Intuitively, this means that two attendees C and D would be considered similar if they share many common words in their associated profiles, and even more so if only a few users share those words. Users have high similarity in set of interests will be clustered into the same community. To protect user privacy, an attendee's interests are stored locally and are not disclosed to other parties including the PCRS. Therefore, a secure multi-party computation mechanism is needed to compute the similarity between every two attendees. We present in the next subsection the similarity calculation procedure in PCF protocol as follows:

1. For any attendee $C, D \in V$ and a set of word vectors $\mathrm{e_c(w_i)}$ and $\mathrm{e_d(w_i)}$, the similarity is calculated in two steps first, it computes the numerator $|V_\mathrm{C} \cap V_\mathrm{D}|$ between attendee C and D and then it computes the denominator $|V_\mathrm{C}|^2+|V_\mathrm{D}|^2$.
2. After selecting a super-peer as the root for computations, a virtual ring topology between attendees is employed for calculating the numerator between every two participants. Each public profile is associated with certain interests that need to be compared with other participants' public profiles then they submit similarity values to super-peers. Both attendees C and D apply a hash function h to each of their word vectors to generate $V_c = h(e_c(w_i))$ and $V_d = h(e_d(w_i))$. EMCP at attendee C generates an encryption E and decryption U keys then it submit the encryption key E to D.
3. Encryption agent at attendee D hides V_d by $B_d = \{e_d(w_i) \times r^D | w_i \in V_d\}$ where r is a random number for each interest w_i, and send B_d to C.
4. Encryption agent at attendee C signs B_d and get the signature S_d, then sends S_d to D again with the same order it receives. EMCP at attendee D reveals set S_d using the set of r values and obtains the real signature SI_d, then it applies hash function h on SI_d to produce $SIH_d = H(SI_d)$.

5. Encryption agent at attendee *C* signs the set V_c and gets signature SI_c then applies same hash function *h* on SI_d to produc e $\mathrm{SIH_c} = \mathrm{H(SI_c)}$ and submits this set to D.
6. Encryption agent at attendee *D* compares SIH_d and SIH_c using the knowledge of V_d, *D* gets the intersection set $IN_{C,D} = SIH_c \cap SIH_d$ that represent $|\mathrm{V_c} \cap \mathrm{V_D}|$. EMCP at *D* applies hash function *h* on $IN_{C,D}$ then it encrypts this value along with $|\mathrm{V_D}|$, $|\mathrm{V_C}|$ and attendees pseudonyms identities using super-peer public key and forwards them to super-peer of this group.
7. Super-peer collects all these results and decrypts them with its private key. Then it starts to cluster participants into communities, such that each community contains participants who share similar interests. Super-peer performs S-seeds [8] clustering algorithm as follows first, randomly select S attendees' profiles as clusters representatives. Then, it calculates the distance between these S seeds and each data point as specified in PCF protocol. Then, assigns each point to the community with the closest seed. Inside each community, choose the point with the smallest average distance to other data as the new seed. Finally, repeat last two steps until the S-seeds do not change. In S-seeds clustering, only the distance calculations among data points are required to identify the communities without disclosing attendees' profiles.

The above protocol performs it computations on *m* hashed values held by *m* parties without exposing any of the inputs values. This protocol is based on secure multi-party computation (SMPC), which was studied first by Yao in his famous Yao's millionaire problem [15].

5.2 Private Sub-Community Discovery (PSD) Protocol

Encryption agent in EMCP executes PSD protocol on the proximate general communities extracted from PCF protocol, PSD protocol determines in a bilateral manor the associated interests within attendees' public profiles, then the final results is used in building sub-communities. PSD protocol is adapted from the work in [16, 17] with the intuition that many frequent interests of attendees should be shared within a sub-community (group) while different sub-communities should have more or less different frequent interests. However, there are no predefined sub-communities yet inside these communities; hence PSD should operate with the available bounded prior domain knowledge and full dimensional profiles.

Definition 2 (Frequent interests) Frequent interests is a notion similar to frequent itemsets in association rule mining, it represent a set of interests that occur together in some minimum fraction of attendees' profiles. For example, let's consider two frequent interests, "libraries" and "C". Profiles containing the interest "libraries" may relate to digital archiving services and Profiles that contain the interest "C" may relate to Healthcare services. However, if both interests occur together in many profiles, then a specific interest sub-community related to C-programming should be identified.

Definition 3 (Global Frequent Interests) Global frequent interests is a set of interests that appear together in more than a minimum fraction of the whole attendees 'profiles in community C; a minimum community support is specified for this purpose. If this set contains k-interests, it called global frequent k-interests such that each interest that belongs to this set is called global frequent interest. Global frequent interest is frequent in sub-community c_i if this interest is contained in some minimum fraction of attendees' profiles; a minimum sub-community support is specified for this purpose.

The attendees are arranged in hierarchical topology in order to compute sub-communities, PSD protocol can be summarized as follows:

1. The initialization process of PSD protocol is invoked by PCRS, whereas attendees form groups then after they negotiate with each other to elect a peer who will act as a "super-peer" for each group. Super-peers distribute a list of 1-candidate frequent interests; therefore, different group members run concurrently a local algorithm to generate local frequent interests using their local support and closure parameters. we use the algorithm presented in [18] to find global & local frequent interests for each group.
2. After local extraction of frequent interests at each member $\forall_1^n \mathrm{P_i}$, member $\mathrm{P_i}$ encrypts this local list with his own key and send it to member $\mathrm{P_{i+1}}$, such that each member successively sends both his local and received lists to next neighbor. Last member in the group $\mathrm{P_{n-1}}$ send collected message to the super-peer. Super-peers now, have a set of local supports and closures of candidate frequent interests; generating global support is done by making the sum of these local supports. The global closure is calculated using intersection of the collected local closure. In the same way, repeating the previous steps, super-peer can generate the candidates of higher size. In order to decrypt the final results, the super-peer encrypts and sends global supports & closures lists to member $\mathrm{P_{n-1}}$ in arbitrary order. Member $\mathrm{P_{n-1}}$ decrypts his encryption from these lists using his own private key, and then sends this list to the next member $\mathrm{P_{n-2}}$ in arbitrary order. When super-peer receives these lists back, these lists will be encrypted with his own key only, which enables him/her to obtain final results.
3. For each adjacent set of global frequent interests at super-peer side, we setup an initial sub-community that includes all attendees' profiles that contain these interests, such that all profiles in this sub-community contain all these global frequent interests. These initial sub-communities are overlapped because each profile may contain multiple global frequent interests. PSD will use these global frequent interests as a sub-community representative. Then after, for each attendee's profile V_i, encryption agent determines the best initial sub-community c_i using the following score function:

$$SimilarityScore(c_i \leftarrow V_i) = \left[\sum\nolimits_{w_i} e_r(w_i) * sub - community\ support(w_i)\right] - \left[\sum\nolimits_{w'_i} e_r\left(w'_i\right) * community\ support\left(w'_i\right)\right]$$

where $\mathrm{w_i}$ is a global frequent interest in profile *r* and this interest is also frequent in sub-community $\mathrm{c_i}$ while $\mathrm{w'_i}$ is a global frequent interest in profile r and is not frequent in sub-community $\mathrm{c_i}$. $\mathrm{e_r(w_i)}$ and $\mathrm{e_r\left(w'_i\right)}$ are the weighted frequency of $\mathrm{w_i}$ and $\mathrm{w'_i}$ in profile *r*, which already calculated during the execution of PCF protocol. After this scoring, each attendee‘s profile belongs to exactly one sub-community.

4. For each community, super-peer organizes sub-communities in hierarchical structure using global frequent k-interests in each sub-community as representatives. In that case, PSD treats all attendees' profiles in each sub-community as single conceptual profile. The sub-community with k-interests will appear at level k in this structure, while the parent sub-community at level k-1 must be a subset of its child sub-community's representatives at level k. The selection of the potential parent for each child sub-community is done using scoring function presented in previous step. After that, super-peers exchange discovered sub-communities with each other to efficiently remove the overly sub-communities based on inter sub-community similarity. The same frequent interests might be distributed over multiple small sub-communities obtained from different super-peers' results, thus merging every two sub-communities into one general sub-community occurs only if they are very similar to each other. Inter sub-community similarity is similar to scoring function presented before with the only difference is that this similarity value should be normalized to remove the effect of varying number of attendees in each sub-community, it is measured using the following functions:

$$SubcommunitySimilarity\left(c_i \leftarrow c_j\right) = \left[SimilarityScore\left(c_i \leftarrow \forall_{x=1}^{n} V_x \in c_j\right) \middle/ \left[\sum\nolimits_{w_j} e\left(w_j\right) + \sum\nolimits_{w'_j} e\left(w'_j\right)\right]\right] + 1$$

Then, $Intersubcommunity\ similarity(c_i \leftrightarrow c_j) = \left[SubcommunitySimilarity(c_i \leftarrow c_j) * SubcommunitySimilarity(c_j \leftarrow c_i)\right]$

where c_i and c_j are two sub-communities; $\forall_{x=1}^{n} V_x \in c_j$ stands for single conceptual profile for sub-community c_j. w_j represents a global frequent interest in both c_i and c_j while w'_j represent a global frequent interest in c_j only but not in c_i. $e(w_j)$ and $e(w'_j)$ are the weighted frequency of w_j and w'_j sub-community c_j

5. Finally, for a new attendee, in order to privately recommend suitable sub-communities for him/her, EMCP obtains a list of sub-communities representatives then it generalizes his/her host interests and extract frequent interests for

this generalized profile. EMCP encrypts these frequent interests and measure their similarity with sub-communities' representatives in order to build a list of similar sub-communities. Finally EMCP assigns his/her host to the sub-community with the highest similarity.

6 Experiments

In this section, we describe the implementation of our proposed solution. The experiments are run on 2 Intel® machines connected on local network, the lead peer is Intel® Core i7 2.2 GHz with 8 GB Ram and the other is Intel® Core 2 Duo™ 2.4 GHz with 2 GB Ram. We used MySQL as data storage for the participants' profiles that is acquired by learning agent. PCRS has been implemented and deployed as a web service while *EMCP* has been deployed as an applet to handles the interactions between its owner, PCRS and other participants; it uses the implementation of the MPI communication standard for distributed memory implementation of our proposed protocols to mimic a distributed reliable network of peers. Our proposed protocols implemented using Java and boundycastel© library, RSA key length is set to 512 for the experimental scenario. The experiments were conducted using a dataset pulled from a recruiter network in Denmark (Manpower Professional) in period of 1990–1997. It contains registration data and information related to different participants that attend exhibitions organized by this agent which held concurrently with various scientific conferences. This data set is comprised of approximately 67,000 users and contains various details about them. Each of those details fell into one of several categories: affiliation, expertise, domains, projects, activities, publication and awards, etc. Due to the lack of a reliable subject authority, some other categories were discarded from all experiments. To generate the public profiles from these profiles we use same method proposed in [8].

In the first experiment, we want to measure the execution time for PCF protocol, from first step to last step at each attendee (excluding the time required to generate RSA keys). We divided our dataset into approximately same number of records and distribute then between 20 participants, then we run this experiment 7 times, so each point in the Fig. 1 is the mean value of the 7 runs. Additionally, we performed two other experiments in our dataset in which data was not divided into parts of same number of records. The first experiment, one client got 60 % of total number of records and the rest of records were divided to other clients as parts of approximately same number of records. While, in the second one, one client got 40 % of total number of records, other clients got the rest. The results of these experiments are summarized in Fig. 1. The results indicate the performance benefits of our protocol, as it is not sensitive to the number of shared interests (Fig. 2).

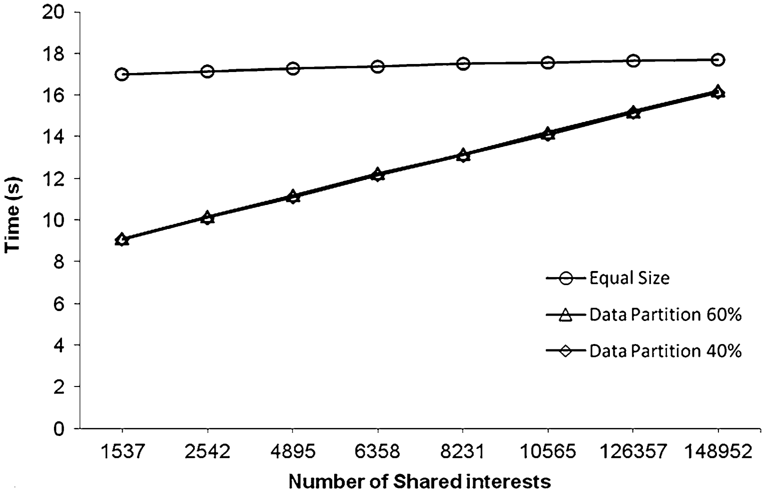
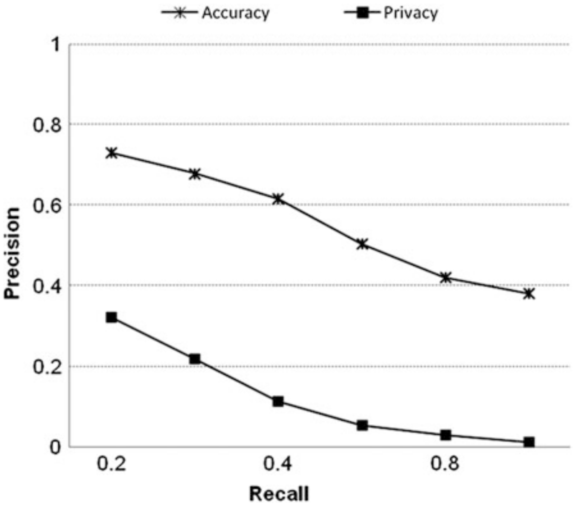


Fig. 1 Execution Time for PCF Protocol

Fig. 2 Recommendations Accuracy and Privacy



In the next experiment, we need to measure the accuracy of extracted sub-communities using PSD protocol. In order to evaluate the accuracy of our results, we apply hierarchical agglomerative clustering in our dataset in order to indentify natural sub-communities from attendees' private profiles. These sub-communities are utilized for measuring the accuracy of the results produced by PSD protocol. Each cluster represents a sub-community which is constructed from a set of attendees' private profiles who share the same specific interests about the same topic. To measure the goodness of our results, we considered two error metrics

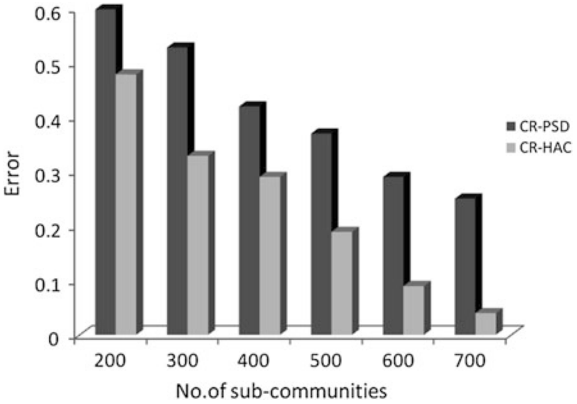


Fig. 3 Grouping Error (GR) of PSD Protocol

defined in [19] which are grouping error (GR) and critical error (CIE). The first one, the grouping error (GR), takes into account the number of attendees' profiles included in a sub-community, but belonging to a topic different from the dominant topic in that sub-community. The second one, the critical error (CIR) measures the number of attendees' profiles belonging to a topic that is not the dominant one in any sub-community. The graphs in Fig. 3 and 4, contain both GR and CIE values for the results obtained from both hierarchical clustering and PSD protocol for different number of sub-communities. This experiment is performed on two versions of our dataset; attendees' generalized profiles are utilized by our PSD protocol, while hierarchical agglomerative clustering utilizes attendees' private profiles that should kept private in our scenario.

We can deduce that both GR and CIE for PSD decrease with the increase in no. of sub-communities till reaching natural number of sub-communities. This indicates that achieving privacy is feasible and does not severely affect the accuracy of the generated sub-communities.

In the last experiment on PSD protocol, we want to measure the overhead of the execution time when applying PSD protocol to preserve attendees' privacy. We divided our dataset into different number of records from 30,000 to 67,000, such that each party held approximately the same number of records. We recorded the execution time when applying our PSD with encryption and without encryption on this data, then we calculated the percentage as following:$\mathit{percentage} = \left({}^{\mathit{time\ without\ encryption}}/_{\mathit{time\ with\ encryption}} \right) * 100$.The graph in Fig. 5 shows time comparison of our PSD protocol with and without encryption for different sizes of our dataset. From the results, we can find that the proposed PSD protocol has a reasonable performance and the privacy preserving nature has marginal impact on the execution time in comparison with non encryption option.

In order to measure the correctness of our solution to capture correlated interests between attendees. We extracted sample data from conference proceeding

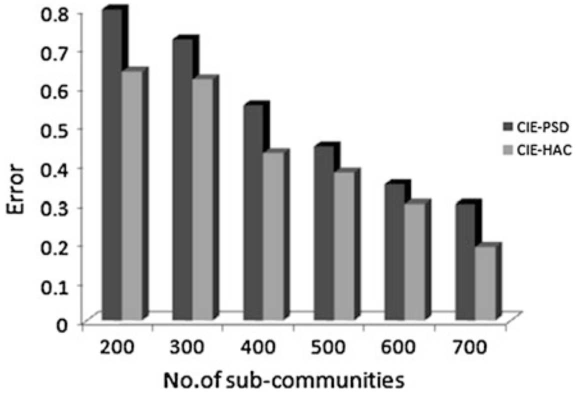


Fig. 4 Critical Error (CIE) of PSD Protocol

related to 500 authors and co-authors. We crawled authors' website to create public profiles for them. Our aim here is to determine if our proposed solution can group attendees in the same sub-community and help them to find the right people to communicate or work with. For every sub-community recommendation for each participant in the conference, we need to test whether or not participants knew each other in this sub-community from previous work and if this recommendation accurate or not. Figure 6 shows a breakdown of the results by our protocols, the percentage of unknown attendees recommended by EMCP are shown above the horizontal center line and the percentages of co-authors below. The chart also shows the percentage of accurate versus inaccurate in two different colors. PCF algorithm recommends other participants than the co-authors, which is not surprising because it mostly creates communities considering only similar interests without take in considerations the correlations between these preferences. In contrast, applying PCF and PSD extract sub-communities for people that are likely similar as sub-communities relies heavily on associations between preferences. These results confirm our intuitions that the more associations between participants' preferences, the more accurate sub-communities are produced.

In the last experiments, we evaluated the proposed solution from different aspects: privacy achieved and accuracy of results. We used precision and recall metrics proposed in [8] to measure privacy and accuracy of the results, the results are shown in Fig. 2. As we can see, a good quality is achieved due to: identifying communities that involve different sub-communities enables accurate recommendations to the attendees who share the same interests. Also, the effect of each interest inside the community can be easily measured, which enables to detect and remove outlier values that are very different than the general interests. We also evaluated the leaked private interests of different attendees when running our solution. We consider users, who published portion of their real interests in their public profiles, for each of these users; we tried the attack procedure proposed in threat model to reveal other hidden interests in their profiles based on the sub-

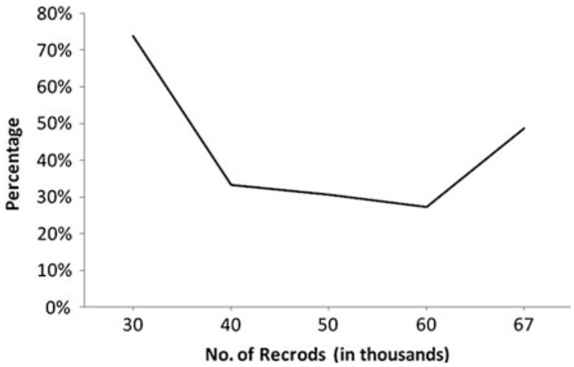


Fig. 5 Percentage Time for PSD Protocol

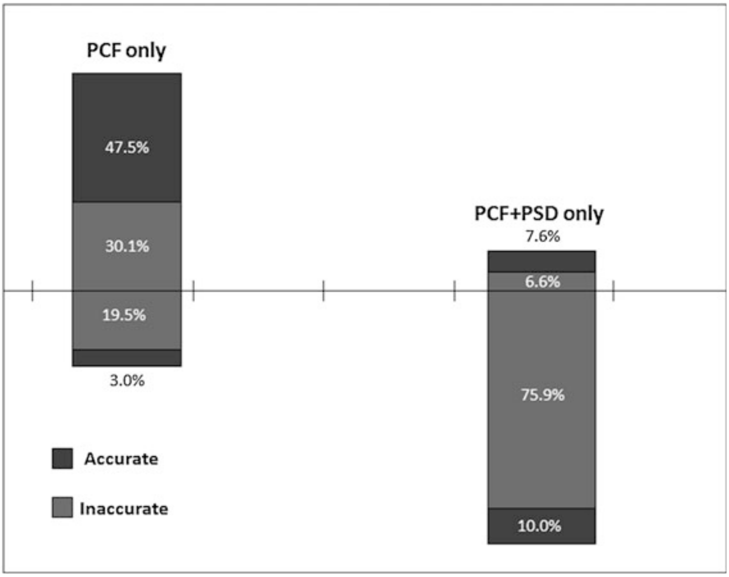


Fig. 6 Co-authors versus Unknown, Accurate versus Inaccurate

community they belong. The obtained interests are quantified using our proposed metrics and the results are shown in Fig. 2. As we can see, our solution manages to reduce privacy leakages for exposed attendees' private interests, However, the revealed interests are only a hashed hypernym terms for attendees private interests.

7 Conclusion and Future Work

In this paper, we presented our attempt to develop an enhanced middleware for collaborative privacy for community based recommender service in conferences or exhibitions. We gave a brief overview of EMCP architecture and proposed protocols. We tested the performance of the proposed protocols on a real dataset. The experimental and analysis results show achieving privacy in recommending subcommunities is feasible under the proposed middleware without hampering the accuracy of the recommendations. A future research agenda will include utilizing game theory to better formulate user groups, sequential preferences release and its impact on privacy of whole profile.

Acknowledgments This work partially supported by the European Comission via the ICT FP7 SOCIETIES Integrated Project (No. 257493). Also it was partially supported from the Higher Education Authority in Ireland under the PRTLI Cycle 4 Programme, in the FutureComm Project (Serving Society: Management of Future Communications Networks and Services).

References

1. He, J., Chu, W.W., Liu, Z.: Inferring privacy information from social networks. Proceedings of the 4th IEEE international conference on Intelligence and security informatics. Springer, San Diego, 154–165 (2006)
2. McSherry, F., Mironov, I.: Differentially private recommender systems: building privacy into the net. Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, Paris, France 627–636 (2009)
3. Esma, A.: Experimental demonstration of a hybrid privacy-preserving recommender system. In: Gilles, B., Jose, M.F., Flavien Serge Mani, O., Zbigniew, R. (eds.), Vol 0 161–170 (2008)
4. Canny, J.: Collaborative filtering with privacy via factor analysis. Proceedings of the 25th annual international ACM SIGIR conference on research and development in information retrieval. ACM, Tampere, Finland 238–245 (2002)
5. Canny, J.: Collaborative filtering with privacy. Proceedings of the 2002 IEEE symposium on security and privacy. IEEE Computer Society 45 (2002)
6. Miller, B.N., Konstan, J.A., Riedl, J.: PocketLens: toward a personal recommender system. ACM Trans. Inf. Syst. **22**, 437–476 (2004)
7. Rubinstein, I.: Regulating privacy by design. Berkeley Technol. Law J., Forthcoming (2011)
8. Elmisery, A., Doolin, K., Botvich, D.: Privacy Aware community based recommender service for conferences attendees. 16th International conference on knowledge-based and Intelligent information & engineering systems. IOS Press, San Sebastian, Spain (2012)
9. Elmisery, A., Botvich, D.: Privacy Aware Recommender Service using Multi-agent Middleware- an IPTV Network Scenario. Informatica 36 (2012)
10. Elmisery, A., Botvich, D.: Enhanced middleware for collaborative privacy in IPTV recommender Services. J. Convergence **2**:10 (2011)
11. Elmisery, A., Botvich, D.: Privacy aware recommender service for IPTV networks. 5th FTRA/IEEE international conference on multimedia and ubiquitous engineering. IEEE, Crete, Greece (2011)
12. Elmisery, A., Botvich, D.: Multi-agent based middleware for protecting privacy in IPTV content recommender services. Multimed Tools Appl 1–27 (2012)

13. Elmisery, A., Botvich, D.: Privacy aware obfuscation middleware for mobile jukebox recommender services. The 11th IFIP conference on e-business, e-service, e-society. IFIP, Kaunas, Lithuania (2011)
14. Sebastiani, F.: Machine learning in automated text categorization. ACM Comput. Surv. **34**, 1–47 (2002)
15. Yao, A.C.: Protocols for secure computations. Proceedings of the 23rd annual symposium on foundations of computer science. IEEE Computer Society 160–164 (1982)
16. Beil, F., Ester, M., Xu, X.: Frequent term-based text clustering. Proceedings of the eighth ACM SIGKDD international conference on knowledge discovery and data mining. ACM, Edmonton, Alberta, Canada 436–442 (2002)
17. Fung B.C.M.: Hierarchical document clustering using frequent item sets. Master's thesis, Simon Fraser University (2002)
18. Cheung, D.W., Han, J., Ng, V.T., Fu, A.W., Fu, Y.: A fast distributed algorithm for mining association rules. Proceedings of the fourth international conference on on parallel and distributed information systems. IEEE Computer Society, Miami Beach, Florida, United States 31–43 (1996)
19. Cuesta-Frau, D., Pérez-Cortés, J.C., Andreu-Garcia, G.: Clustering of electrocardiograph signals in computer-aided Holter analysis. Comput. Methods Programs Biomed. **72**, 179–196 (2003)