International Journal of Scientific Research in Computer Science, Engineering and Information Technology
© 2017 IJSRCSEIT | Volume 2 | Issue 6 | ISSN : 2456-3307

Protected and Effective Profile Matching in Mobile Social Networks

Gurunadham R[1], Redapangu Saidulu[2]
Assistant Professor[1], M.Tech Scholar[2]
Department of Computer Science and Engineering, CVSR College of Engineering, Venkatapur, Telangana, India

ABSTRACT

Recently, *mobile social networks (MSNs)*have been widely discussed due to the rapid growth of smart mobile devices. This work focuses on *mobile D2D social networks (MDSNs)*, where usersin an MSN are physical neighbors. An important social application of MDSNs is *common profilematching (CPM)*, which refers to the scenario wherea group of smart phone users meet in a small region (such as a ball room) and these users are interested in identifying the common attributes among them from their personal profiles efficiently via short-range (such as D2D) communications. For example, a group of strangers may want to find common hobbies, friends, or countries they visited before, and a group of students may want to know the common courses they have ever taken. Assuming that users in an MDSN form a fully connected network, we formulate three versions, namely *all*-common, _-common, and top--popular, of the CPM problem. The first problem is an extension of an earlier work, while the latter two problems are newly defined. We present solutions based on the *basic* and the *iterativeBloom filters*. Evaluation results show that ourmechanisms are quite communication-efficient.

Keywords : CPM, MSN, Iterative Bloom Filters

I. INTRODUCTION

As online social networks and Twitter-like micro-blogging services redefine our lifestyle, groups are becoming one of the most frequently used features. Groups are, in general, formed with common attributes, such as geographic locations and hobbies. However, the features of a group are generally described by only a few keywords or a short description, which sometimes is not enough for users to make decisions when choosing an appropriate group for themselves. Especially, when several groups have similar (or even the same) keywords and descriptions, it is very inconvenient for users to choose the most suitable one among these groups. In order to make a better decision when choosing a group to join, a stranger with a profile of his own attributes — who is still an outsider of the group —needs to collect detail matching information from all the group members' profiles. Such a problem is referred as to *group matching*.

In most situations, attributes of users are sensitive, such as personal health records and religious preferences. It is typical for a user to store these attributes privately [1], so that only his friends or members in the same group are able to reveal these attributes, but strangers or any third party cannot learn this sensitive information. Unfortunately, collecting group matching information using these sensitive attributes may introduce a number of privacy problems. On one hand, since the stranger is not familiar with the group, the stranger does not want to reveal his sensitive attributes to any group member during the matching process. On the other hand, because the stranger is an outside and UN trusted user to the group, each group member is reluctant to reveal his own attributes and the exact matching results between two entities to the stranger. To make matters more challenging, each group member needs to generate a signature on his matching response, which contains matching information between the stranger and himself, and sends the signature and the matching response together to the stranger, so that the stranger is convinced the matching response is reliable and correct. Unfortunately, due to the *un forge ability* of signatures (only the entity with the knowledge of the private key can create valid signatures), the stranger is

able to learn the identity of the signer on each matching response, and reveal exact matching information between himself and each group member.

In this paper, we proposed G match, a novel secure and privacy-preserving group matching scheme in online social networks. We utilize private set intersection [2] in G match, so that the stranger is able to collect matching information from the group while both the stranger and each group member are able to preserve sensitive attributes to each other.
Meanwhile, with ring signatures [3], [4], the stranger is convinced that matching information from the group is correct, but he cannot learn exact matching information between himself and each group member. In addition, we improve the efficiency of the matching process using batch verification.

II. GMATCH : SECUREAND PRIVACY-PRESERVING GROUP MATCHING

G match includes four steps: ***Setup***, Compute, ***Evaluate***, ***and Match***. In ***Setup***, stranger S and eachgroup member generate their own public/private key pairs. In ***Compute***, stranger S first Generates a polynomial, where each attributes in his profile is a root of this polynomial and all the roots are in his profile. Then, stranger S encrypts all the coefficients of this polynomial by performing additive homomorphism encryption, and sends all the encrypted coefficients to all the group members. In ***Evaluate***, each group member evaluates a matchingvalue for each attribute in his own profile using all the encrypted coefficients, signs a matching response that contains all the matching values generated by him, and sends this matching response and the corresponding signature to the stranger. In ***Match***, stranger S first checks the correctness of a matching response by verifying its signature, and then computes whether each matching value in this matching response indicates a matched attribute. After collecting all the matching responses from all group members, the stranger S calculates matching degrees for all the attributes in his profile. Details of each step are listed as follows.

Setup Stranger S generates his public/private key pair(pk_s, sk_s) for additive homomorphism encryption. Here, we utilize Parlier cryptosystem [8]. The encryption algorithm is denoted as Enc, and the corresponding decryption algorithm is denoted as Dec. Each group member generates his public/private key pair (pk_i, sk_i) for computing ring signatures. The ring signature scheme we used is BGLS [4], which is based on bilinear maps. The total number of group members is d. The number of attributes in the stranger's profile is k, and the number of attributes in each group member's profile is m.

Algorithm 1 KeyGen

Given two multiplicative cyclic groups G_1, G_2 with prime order p and their generators g_1, g_2 respectively, group member P_i generates his public key and private key as:

1) Pick random $u_i \in Z_p$.
2) Compute $v_i = g_2^{u_i} \in G_2$.

Group member P_i's public key is $pk_i = v_i$ and his private key is $sk_i = u_i$.

Compute.Stranger S first constructs a k-degreepolynomial P(x), whose k roots are all attributes in his profile. This polynomial is described as:

$$P(x) = (x - a_{s,1})(x - a_{s,2}) \dots (x - a_{s,k}) = \sum_{i=0}^{\kappa} \alpha_i x^i. \quad (1)$$

Clearly, if an attribute $a_{i,j}$ from group member Pi is a matched attribute that equals some attribute in stranger S's profile, then $a_{i,j}$ is also a root of this k-degree polynomial P(x), and we have $P(a_{i,j}) = 0$.

After generating polynomial P(x), stranger S encrypts all the k+1 coefficients of this polynomial P(x) using Enc with his public key pk_s. He then sends all the k + 1 encrypted coefficients {Enc (∝0)... Enc (∝k)} to each group member (as illustrated in Fig.1).

Evaluate.Group member $P_i$has m attributes andevaluates a matching value $w_{i,j}$ for each attribute $a_{i,j}$ in his profile. More specifically, group member P_i first computes an encrypted polynomial value Enc (P ($a_{i,j}$)) for each attribute

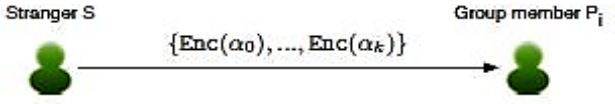

Figure 1. Stranger S sends all the encrypted coefficientsto group member Pi.

Algorithm 2 RingSign

Given all the group members' public keys $(pk_1, ..., pk_d) = (v_1, ..., v_d)$, a matching response $\boldsymbol{w}$, and a private key $sk_s = u_s$ for some s, this group member u_s

1) Randomly chooses $y_i \in Z_p$ and computes $\sigma_i = g_1^{y_i}$ for all $i \neq s$ and $i \in [1, d]$.
2) Computes $h = H(\boldsymbol{w}) \in G_1$ and sets

$$\sigma_s = \left(\frac{h}{\psi(\prod_{i \neq s} v_i^{y_i})} \right)^{1/u_s}, \quad (4)$$

where $H : \{0,1\}^* \rightarrow Z_p$ is a full-domain hash function and $\psi : G_2 \rightarrow G_1$ is a computable isomorphism.

3) Outputs the ring signature $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_d) \in G_1^d$.

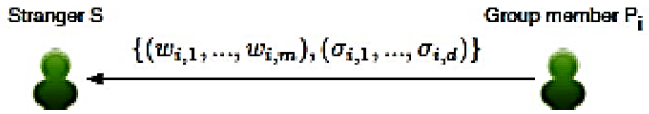

Figure 2 : Group member Pi sends matching response $w_i$and its signature _i to stranger S

Match. Upon receiving a matching response w_i andits ring signature _i, stranger S first verifies the correctness of this matching response according to Algorithm 3. If the matching response passes the verification, stranger S decrypts each $w_{i,j}$_w_i with decryption algorithm Dec. If the result of decryption matches one of his attributes, then $a_{i,j}$ is a matched attribute. Otherwise, it is an unmatched attribute. This is because

$$\begin{aligned} \mathrm{Dec}(w_{i,j}) &= \mathrm{Dec}(\mathrm{Enc}(\tau_{i,j} \cdot P(a_{i,j}) + a_{i,j})) \\ &= \tau_{i,j} \cdot P(a_{i,j}) + a_{i,j}, \end{aligned} \quad (5)$$

where $P(a_{i,j}) = 0$ and $\mathrm{Dec}(w_{i,j}) = a_{i,j}$, if $a_{i,j} \in \mathcal{A}_s$.

Algorithm 3 RingVerify

Given all the group members' public keys $(pk_1, ..., pk_d) = (v_1, ..., v_d)$, a matching response $\boldsymbol{w}$, and its ring signature $\boldsymbol{\sigma} = (\sigma_1, ..., \sigma_d)$, the stranger

1) Computes $h = H(\boldsymbol{w}) \in G_1$.
2) Verifies

$$e(h, g_2) \stackrel{?}{=} \prod_{i=1}^{d} e(\sigma_i, v_i). \quad (6)$$

If the equation holds, then this matching response is correct and signed by a group member. Otherwise, it is not.

After decrypting all the matching values from all the group members, stranger S is able to calculate the matching degree D_j , for j _ [1, k] and obtain group matching information D(P) = (D1, ...,D k) about this group P.

III. COMMON PROFILE MATCHING IN AN MDSN

A. System Model

In an MDSN, we consider one user group U and one *attribute profile* of these users. (An attribute profile could be the hobbies, the countries visited, the courses taken, or the celebrities followed by users. Solving this problem for multiple user groups and attribute profiles can be done by repeating this for each user group and attribute profile.) Let U = {u1, u_2. . . u_q} and the attribute profile of user u_i_ U be P_i, i = 1..q. Each element in P_i is called an *attribute item*. Let $n_i = |P_i|$. The universal set of the attribute items is denoted by P and we assume that P is known by all users.

We propose three versions of the CPM problem.

• *All - common*: The goal is to find the intersection of all users' attribute profiles,
i.e., $C_{all} = \cap i=1...q\ P_i$.

• _-common: The goal is to find the set of all attribute items such that each attribute item is in at least _ users'

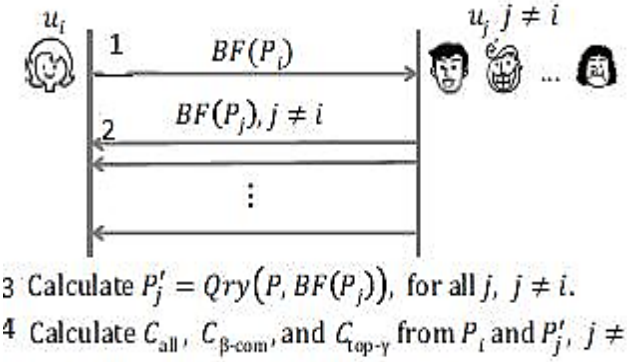

Figure 3 : Workflow of the basic Bloom filtersolution.

Attribute profiles. For any a ∈ P, we define a membership function _(a, Pi) such that _(a, Pi) = 1 if a ∈ Pi and _(a, Pi) = 0 otherwise. The
common set is defined as C = {a|_

=1...q_(a, Pi) ≥ _}.

• top--popular: The goal is to find the set of top-hottest attribute items that are shared by all users' profiles. The top- set is defined as C_{top-} = {a| the value _ =1..q_(a, Pi) of a is ranked top-}.

Basic Bloom Filter Solution

We present a basic Bloom filter solution for solving the CPM problems. Assuming that users in U form a fully connected network, our solution first requests each user u_i, i = 1..q, to insert her attribute profile Pi into a Bloom filterand sends the m-bit array to all other users in U.

Iterative Bloom Filter Solution

This solution is based on the iterative Bloom filter (IBF) [19]. The goal is to further reduce the message cost by using smaller Bloom filters iteratively. Recall that the size of a Bloom filter should be set according to the expected number of inserted items and the expected false positive rate. We can use a smaller Bloom filter (thus with a higher false positive rate) in the first iteration to reduce message cost. Since some items have been filtered out, in the subsequent iterations, we can still use smaller Bloom filters while controlling the false positive rate. Therefore, IBF may keep the same false positive rate as the basic solution with smaller message cost. In [19], a 2-iteration IBF has been shown to be quite efficient. Hence, we adopt the same approach. The following steps are executed by each user u_i, i = 1...q, concurrently. Note that the value of m is not necessarily the same as that used in the basic Bloom filter solution. (Refer to Fig. 4 for illustration.)

IV. RELATED WORK

A. Two-party private matching

Freedman *et al.* [2] proposed a private matching scheme, which allows a client and a server compute the set intersection with their own private sets. During private matching, the client only obtains the set intersection while the server does not know any matching result. Agrawal*etal.* [6] introduced a private matching schemebetween two databases using commutative encryptions. Hazay and Lindell [7] exploited pseudo random functions to evaluate set intersection. In [11], Dachman-Soled *et al.* exploited polynomial evaluations to compute theset intersection between two parties, and also leveraged Shamir secret sharing and cut-and-choose protocol to improve efficiency. Recent work in [12] introduced an authorized private Set intersection (APSI) based on blind RSA signatures. In APSI, each element in the client's set must be authorized by some mutually trusted authority.

B. Multi-party private matching

Kissner and Song [13] proposed a multi-party private matching scheme to compute the union, intersection and element reduction operations for multiple sets. However, this scheme requires a group decryption among multiple entities, which is impractical between the stranger and group members in social networks. Ye *et al.* [14] extended previous scheme to a

Distributed scenario with multiple servers The dataset of the original server is shared by several sub-servers using (t,w) - shamir secret sharing. Therefore, any t−1 or fewer sub-servers cannot discover the dataset of the original server. Sang *et al.* [15] improved the efficiency of privatematching among multiple parties by exploiting an extra N × N nonsingular matrix, where N is the total number of entities. Li and Wu [10] proposed a private multi-party set intersection scheme based on the two-dimensional verifiable secret sharing scheme.

C. Private matching in social networks

Find U [1] focuses on finding the *best matched* user from the group in mobile social networks. Yang *et al.* [16] introduced E-Small Talker, which allows users to privately match other people in mobile social networks using the iterative bloom filter (IBF) protocol.

V. CONCLUSIONS

In this work, we have presented CPM in an MDSN, which is to identify common attributes among physical neighbors via D2D communications. We consider three CPM problems: *all*-common, _-common, and top--popular. We propose basic and iterative Bloom filter-based solutions to these problems. We have conducted some evaluations on commercial smart phones to prove that the proposed solutions are communication-efficient. The basic Bloom filter and IBF Solutions perform well in execution time when the number of attribute items in each profile is increased. However, the IBF solution incurs lower message costs. We have also demonstrated a prototype on Android smart phones. The application verifies that theproposed solutions are feasible on off-the-shelf smart phones.

VI. REFERENCES

[1]. M. Li, N. Cao, S. Yu, and W. Lou, "FindU: Private-Preserving Personal Profile Matching in Mobile Social Networks," in Proc. IEEEINFOCOM, 2011, pp. 2435 - 2443.

[2]. M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Matching and Set Intersection," in Proc. EUROCRYPT Spring-Verlag, 2004, pp. 1-19.

[3]. R. L. Rivest, A. Shamir, and Y. Tauman, "How to Leak a Secret," in Proc. ASIACRYPT. Springer-Verlag, 2001, pp. 552-565.

[4]. D. Boneh, C. Gentry, B. Lynn, and H. Shacham, "Aggregate and Verifiably Encrypted Signatures from Bilinear Maps," in Proc.EUROCRYPT. Springer-Verlag, 2003, pp. 416-432.

[5]. A. Sorniotti and R. Molva, "Secret Interest Groups (SIGs) in Social Networks with an Implementation on Facebook," in Proc. ACMSAC, 2010, pp. 621-628.

[6]. R. Agrawal, A. Evfimievski, and R. Srikant, "Information Sharing Across Private Databases," in Proc. ACM SIGMOD, 2003, pp. 86-97.

[7]. C. Hazay and Y. Lindell, "Efficient Protocols for Set Intersection and Pattern Matching with Security against Malicious and Covert Adversaries," in Proc. TCC Springer-Verlag, 2008, pp. 155-175.

[8]. P. Paillier, "Public-Key Cryptosystems Based on Composite Degree Residuosity Classes," in Proc. EUROCRYPT. Springer-Verlag, 1999, pp.223-238.

[9]. C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," in Proc.IEEE INFOCOM, 2010, pp. 525-533

[10]. R. Li and C. Wu, "An Unconditionally Secure Protocol for Multi-Party Set Intersection," in Proc. ACNS. Springer-Verlag, 2007, pp. 226-236.

[11]. D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung, "Efficient Robust Private Set Intersection," in Proc. International Conferenceon Applied Cryptography and Network Security.Springer-Verlag, 2009, pp. 125-142.

[12]. E. D. Cristofaro and G. Tsudik, "Practical Private Set Intersection Protocols with Linear Complexity," in Proc. FC, 2010, pp. 143-159.

[13]. L. Kissner and D. Song, "Privacy-Preserving Set Operations," in Proc.CRYPT. Springer-Verlag, 2005, pp. 241-257.