A Novel Privacy-Preserving Group Matching Scheme in Social Networks

Jialin Chi[1,2], Zhiquan Lv[1,2], Min Zhang[1], Hao Li[1], Cheng Hong[1], and Dengguo Feng[1]

[1] Trusted Computing and Information Assurance Laboratory, Institute of Software, Chinese Academy of Sciences, Beijing, China
[2] University of Chinese Academy of Sciences, Beijing, China
{chijialin,lvzhiquan,mzhang,lihao,hongcheng,feng}@tca.iscas.ac.cn

Abstract. The group service allowing users with common attributes to make new connections and share information has been a crucial service in social networks. In order to determine which group is more suitable to join, a stranger outside of the groups needs to collect profile information of group members. When a stranger applies to join one group, each group member also wants to learn more about the stranger to decide whether to agree to the application. In addition, users' profiles may contain private information and they don't want to disclose them to strangers. In this paper, by utilizing private set intersection (PSI) and a semi-trusted third party, we propose a group matching scheme which helps users to make better decisions without revealing personal information. We provide security proof and performance evaluation on our scheme, and show that our system is efficient and practical to be used in mobile social networks.

Keywords: Social Networks, Group Matching, Private Set Intersection.

1 Introduction

Social networks are changing our lifestyle and becoming an inseparable part of our daily lives. For example, Twitter [1] which is a well-known micro-blogging site enables users to share real-time information. The group service has been frequently used in social networks and allowed strangers with similar profiles to construct new relationships and share information. Generally, groups are consisted of users with common attributes, such as educational backgrounds and illness symptoms. In many situations, a group is only described by its classification, several keywords and a short introduction. These features may not be enough for users to decide which group is the most appropriate to join, especially when a few groups have similar keywords and introductions. In order to choose a suitable group to join, a stranger outside of the groups needs to collect profile information about each group member. In addition, attributes of users sometimes contain sensitive and private information, thus they don't want to disclose their profiles or exact matching results to untrusted users or any third party. Such a problem is referred to as *group matching* by Wang *et al.* in [2]. However, there are two main problems in existing works for group matching problem.

F. Li et al. (Eds.): WAIM 2014, LNCS 8485, pp. 336–347, 2014.

The first one is that only the stranger obtains the matching results while each group member learns nothing. In most practical applications, when a stranger who is an outsider of an existing group applies to join it, he just needs to simply send the reasons for application to the manager of this group. Since the reasons submitted by stranger may be incomplete or fake, it is inconvenient for the group manager to determine whether to agree to the application. In addition, sometimes other group members don't fully trust the group manager and they want to make their own decisions. In order to enable all group members to participate in the decision process and make a better decision jointly, each group member needs to learn more information about stranger's profile.

Another problem is that existing systems rely mostly on exponential operations and have high computation cost, so they are not lightweight and practical enough to be used in mobile social networks. The proliferation of networked portable devices such as smart phones and PADs, enables people to use social networking services anytime and anywhere. However, networked portable devices have limited computational abilities, and we have to consider computation cost in mobile social networks.

Our Contribution. In this paper, we focus on the above problems and propose a novel scheme to realize group matching by utilizing private set intersection (PSI) [3] and a semi-trusted third party. Our contributions can be summarized as follows:

- Our scheme helps both stranger and each group member to make better decisions. We take advantage of two kinds of matching information learnt by the stranger and each group member respectively: the intersection set between their attribute sets, and the size of their intersection set. By collecting different kinds of matching information, the stranger can make a better decision when choosing a suitable group to join and each group member can decide whether to agree to the stranger's application.
- We limit the risk of privacy exposure and only necessary information of each user's profile is exchanged. Our system protects each participator's private attributes and exact matching information between two entities. We provide thorough security analysis that our proposed scheme is secure under honest-but-curious (HBC) model and against several certain active attacks.
- We utilize a semi-trusted third party to improve the computation efficiency and our proposed scheme relies mostly on modular multiplication. We provide performance evaluation on our scheme. By comparison with an existing work, we show that ours is much more lightweight and efficient in computation to be used in mobile social networks.

Organization. The remainder of this paper is organized as follows. In Section 2, we discuss the related works. In Section 3, we present the system model and design goals. Section 4 describes the details of our scheme. We give the thorough security proof in Section 5 and analyze the efficiency of our scheme by

comparison with an existing work in Section 6. Finally, we briefly conclude this paper in Section 7.

2 Related Work

Existing works related to our proposed scheme are mainly in the area of private set intersection (PSI) first introduced by Freedman*et al.* in [3]. Freedman *et al.* base their protocol on oblivious polynomial evaluation and the protocol is single-output, i.e., during the process, only one party learns the set intersection while the other one doesn't obtain any results. There have been other single-output PSI protocols. Based on oblivious polynomial evaluation, Dachman-Soled *et al.* [4] present an efficient two-party protocol which is robust in the presence of malicious adversaries. In [5], Hazay and Lindell claim a different protocol based on oblivious pseudo random functions and the proposed protocol is improved in complexity by Jarecki and Liu [6]. Cristofaro and Tsudik [7] propose protocols for plain and authorized private set intersection (PSI and APSI) and they base their protocols on blind RSA signatures. In [8], Agrawal *et al.* adopt another approach based on commutative encryption to realize private set intersection, which is extended by Vaidya *et al.* [9] to multiparty setting.

Above single-output protocols only allow one user to obtain the results, while in most situations, both of the two parties are desirable to learn the intersection of their attribute sets. Several mutual PSI protocols have been proposed. Kissner and Song exploit the first mutual PSI protocol in [10]. The proposed protocol builds upon oblivious polynomial evaluation and enables several set operations such as union, intersection, and element reduction operations. Camenisch and Zaverucha [11] have applied certified sets to private set intersection problem and ensured that all inputs are valid and bound to each protocol participant by utilizing a trusted third party. In [12], Kim *et al.* claim a more efficient mutual PSI scheme which is the first system with linear computational complexity in semi-honest model. Recent work in [13], Dong *et al.* present the first fair mutual PSI protocol by utilizing an offline semi-trusted third party arbiter which can resolve disputes blindly without obtaining any sensitive information from users. However, these mutual protocols can't be utilized in group matching problem directly. Users from the same group are familiar, and a group member may exchange the intersection between him and stranger with other group members to learn more about the stranger's private attributes. In addition, above protocols reveal the exact matching information which is undesirable in our work.

Based on private set intersection (PSI), there have been several practical systems designed for special purposes in social networks. The E-SmallTalker scheme [14] exploited by Yang *et al.* adopts iterative bloom filter (IBF) to denote attribute sets and enables a user to match people in physical proximity. Lu *et al.* [15] present a secure symptoms matching protocol by utilizing a trusted authority. The FindU scheme [16] proposed by Li *et al.* allows a user to find the one who best matches with him in mobile social networks. The proposed protocol is based on the FNP scheme [3], but they utilize secret sharing to calculate polynomial evaluation without using additive homomorphic encryption. Recently in

[2], Wang *et al.* introduce Gmatch that allows a user to find the most appropriate group to join without disclosing each user's private information and exact matching results. In the Gmatch system, only the stranger outside of the groups obtains the matching results while each group member learns nothing and the proposed scheme relies mostly on exponentiation operations.

3 Problem Definition

3.1 System Model

Our system is a mobile social network consisting of a stranger S, a group $\mathcal{P}$ and a semi-trusted third party C, and each user processes a networked portable device such as smart phones and PADs (as illustrated in Fig. 1). The stranger S, who launches the matching procedure, is an outsider of group $\mathcal{P}$ and has n attributes in his profile which is denoted as $\mathcal{A}_s = \{a_{s,1}, \ldots, a_{s,n}\}$. The group $\mathcal{P}$ has d group members $P_1, \ldots, P_d$ and P_i has m attributes in his profile which is denoted as $\mathcal{A}_i = \{a_{i,1}, \ldots, a_{i,m}\}$. For simplicity, we assume each group member has the same size of attribute set, i.e., $|\mathcal{A}_i| = m, 1 \leq i \leq d$. All attributes of every user's profile need to be kept private, and they are stored in local portable devices by each user. The semi-trusted third party C is a computation center with high computational ability to help users complete the matching process, but it doesn't access and collect each user's attributes.

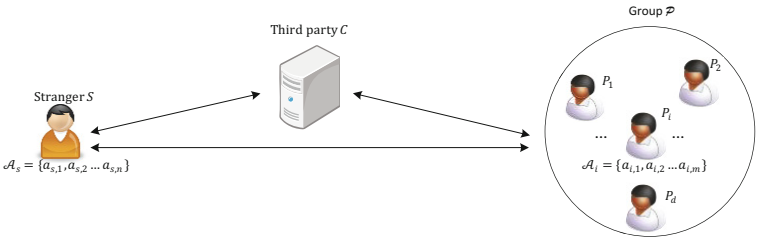

Fig. 1. In order to make better decisions, both stranger and each group member need to learn detail matching information between them

During the matching procedure, stranger S wants to collect the intersection set between him and each group member in order to decide whether group $\mathcal{P}$ is suitable and appropriate to join. When S applies to join $\mathcal{P}$, each group member in $\mathcal{P}$ wishes to learn the size of intersection set between him and stranger S to determine whether to agree to S's application. In this paper, we cite the definition of *matched attribute* and *matching degree* used in [2]. If an attribute in a group member's attribute set is also in stranger S's attribute set, it is called a matched attribute. Otherwise, it is an unmatched attribute. The total number of group members, who has the attribute equal to the attribute $a_{s,j}$ in stranger S's profile, is described as the matching degree D_j of $a_{s,j}$. The matching result

learnt by stranger S can be denoted by the matching degree between S and $\mathcal{P}$, which is described as $\mathcal{D}(\mathcal{P}) = \{D_1, \ldots, D_n\}$.

3.2 Adversary Model

In this paper, we only consider attacks from insiders who are participators of the matching process. We assume all participators including stranger S, group members $P_1, \ldots, P_d$ and the third party C are honest-but-curious (HBC). That means all parties will honestly follow the scheme, but may try to obtain more information than allowed. We will prove our protocol's security under HBC model. We also consider several certain active attacks and analyze how the proposed scheme is secure against them. In addition, we assume that users from the same group are familiar and they may exchange information to learn more about stranger's private profile, while stranger S or any group member can't collude with the semi-trusted third party C.

3.3 Design Goals

Security Goals

Definition 1 (Security Goal 1 (SG-1)): When the scheme ends, stranger S only learns the matching degree $\mathcal{D}(\mathcal{P}) = \{D_1, \ldots, D_n\}$ from group $\mathcal{P}$ without knowing any unmatched attribute of group members and the exact matching information, i.e., each result's corresponding group member and whether two results are from the same user.
Definition 2 (Security Goal 2 (SG-2)): If stranger S doesn't apply to join the group $\mathcal{P}$, each group member in $\mathcal{P}$ will learn nothing about S's attributes, including the intersection set between them and the size of it. If S applies to join, each group member will only learn the size of the intersection set between him and S without knowing what the exact matching attributes are.
Definition 3 (Security Goal 3 (SG-3)): In any phase of our scheme, the semi-trusted third party C can't learn more than what can be derived from the values sent to him, his outputs and their corresponding group members.

Usability and Efficiency. For group matching in mobile social networks, it is better to require as few human interactions as possible. In this paper, stranger S only needs to determine which group is the most suitable and whether to join it, while group members in $\mathcal{P}$ need to decide whether to accede to S's application. In addition, networked portable devices have limited computational abilities, and our scheme should be lightweight and efficient enough in computation to be used in mobile social networks.

4 A Novel Privacy-Preserving Group Matching Scheme

In this section, we propose a novel scheme designed for group matching in social networks. The proposed scheme is based on the FNP protocol [3] and we take

advantage of a semi-trusted third party C to help compute the polynomial evaluations without using additive homomorphic encryption. Our scheme consists of four phases: *Setup*, *Computation*, *Matching* and *Application*. The *Application* phase is only executed when stranger S applies to join the group $\mathcal{P}$. We assume that each party has a public/private key pair for secure communication and the encryption algorithms are denoted as $Enc_c, Enc_s, Enc_1, \ldots, Enc_d$. At first, all attributes in each user's profile are encoded in $\mathbb{Z}_p$. Details of each phase are listed as follows.

Setup. Stranger S first constructs a n-degree polynomial $f(x)$, whose n roots are all in his set of attributes and all his attributes are $f(x)$'s roots:

$$f(x) = (x - a_{s,1})(x - a_{s,2}) \ldots (x - a_{s,n}) = \sum_{k=0}^{n} \alpha_k x^k. \tag{1}$$

After generating the polynomial $f(x)$, stranger S generates $\{r_{i,j}\}_{1 \le j \le m}$ and $\{\tau_{i,k}\}_{1 \le k \le n}$ randomly from $\mathbb{Z}_p$ for each group member $P_i \in \mathcal{P}$. Then he sends the encrypted values $\{Enc_c(\tau_{i,1} r_{i,j} \alpha_1), \ldots, Enc_c(\tau_{i,n} r_{i,j} \alpha_n)\}_{1 \le j \le m}$ to the semi-trusted third party C and $\{Enc_i(r_{i,j}\alpha_0)\}_{1 \le j \le m}$, $\{Enc_i(\tau_{i,k})\}_{1 \le k \le n}$ to group member P_i.

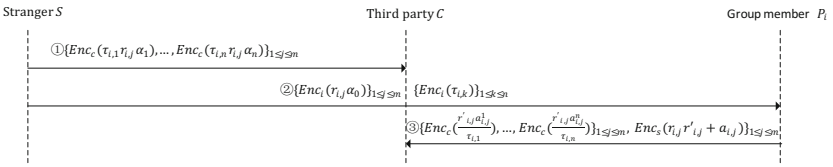

Fig. 2. Stranger S and group member P_i send the parameters used to calculate the matching results to third party C

Upon receiving the values $\{Enc_i(r_{i,j}\alpha_0)\}_{1 \le j \le m}$, $\{Enc_i(\tau_{i,k})\}_{1 \le k \le n}$ and decrypting them with the private key, group member P_i generates $\{r'_{i,j}\}_{1 \le j \le m}$ randomly from $\mathbb{Z}_p$. Then P_i sends $\{Enc_c(\frac{r'_{i,j} a^1_{i,j}}{\tau_{i,1}}), \ldots, Enc_c(\frac{r'_{i,j} a^n_{i,j}}{\tau_{i,n}})\}_{1 \le j \le m}$ and $Enc_s(r_{i,j} r'_{i,j} \alpha_0 + a_{i,j})_{1 \le j \le m}$ to the semi-trusted third party C (as illustrated in Fig. 2).

Computation. After decrypting the received values with his own private key, the third party C now learns $\{\tau_{i,1} r_{i,j} \alpha_1, \ldots, \tau_{i,n} r_{i,j} \alpha_n\}$ and $\{\frac{r'_{ij} a^1_{ij}}{\tau_{i,1}}, \ldots, \frac{r'_{ij} a^n_{ij}}{\tau_{i,n}}\}$ for each attribute $a_{i,j}$ in P_i's profile. C first calculates the intermediate result

$$\begin{aligned} z_{i,j} =& \sum_{k=1}^{n} (\tau_{i,k} r_{i,j} \alpha_k)(\frac{r'_{i,j} a^k_{i,j}}{\tau_{i,k}}) \\ =& r_{i,j} r'_{i,j} f(a_{i,j}) - r_{i,j} r'_{i,j} \alpha_0, \end{aligned} \tag{2}$$

and encrypts it with stranger S's public key. Then C packages the encrypted intermediate result $Enc_s(z_{i,j})$ and its corresponding $Enc_s(r_{i,j}r'_{i,j}\alpha_0 + a_{i,j})$, and sends all the packages $\{Enc_s(z_{i,j}), Enc_s(r_{i,j}r'_{i,j}\alpha_0 + a_{i,j})\}_{1\leq j\leq m}$ to stranger S in random order.

Matching. Upon receiving the packages from the semi-trusted third party C, stranger S decrypts each $Enc_s(z_{i,j})$ and $Enc_s(r_{ij}r'_{ij}\alpha_0 + a_{ij})$ with his private key and computes

$$F_{i,j} = z_{i,j} + r_{i,j}r'_{i,j}\alpha_0 + a_{i,j}. \tag{3}$$

Because the value $f(a_{i,j})$ is randomized by random numbers $r_{i,j}$ and $r'_{i,j}$ generated by S and P_i respectively in *Setup* phase, stranger S will get an attribute in his profile or a random number from the result $F_{i,j}$. If value $F_{i,j}$ is equal to one attribute $a_{s,k}$ in S's profile, $a_{i,j}$ represents a matched attribute which equals $a_{s,k}$. Otherwise, $a_{i,j}$ is an unmatched attribute. Obviously, if $a_{i,j}$ is a matched attribute, it is a root of polynomial $f(x)$, i.e., $f(a_{i,j}) = 0$. Then $F_{i,j} = r_{i,j}r'_{i,j}f(a_{i,j}) + a_{i,j} = a_{i,j}$.

Since stranger S and group member P_i jointly randomize the value α_0 by generating $r_{i,j}$ and $r'_{i,j}$ respectively, and the results are sent by third party C in random order, S won't learn $F_{i,j}$'s corresponding group member and whether two results are from the same user. We also utilize blinding factors $\{\tau_{i,k}\}_{1\leq k\leq n}$ to blind the parameters to compute functions $\{\sum_{k=1}^{n}(r_{i,j}\alpha_k)(r'_{i,j}a^k_{i,j})\}_{1\leq j\leq m}$. Thus the semi-trusted third party C can calculate the correct intermediate results without learning more than what can be derived from the values sent to him, his outputs and their corresponding group members. In our scheme, the value $Enc_s(r_{i,j}r'_{i,j}\alpha_0 + a_{i,j})$ can't be sent to stranger S by P_i directly, otherwise S won't know its corresponding intermediate result.

After computing all results $\{F_{i,j}\}_{1\leq i\leq d,1\leq j\leq m}$ and comparing them with his own attributes, S learns each attribute $a_{s,k}$'s matching degree D_k and decides whether to join group $\mathcal{P}$. If stranger S determines to join it, the *Application* phase will be executed. Otherwise, the matching procedure is done.

Application. Stranger S first generates $\{\omega_{i,j}\}_{1\leq j\leq m}$ randomly from $\mathbb{Z}_p$ for each group member P_i. Then he calculates $\{r_{i,j}\alpha_0 - \omega_{i,j}\}_{1\leq j\leq m}$ and sends $\{Enc_c(r_{i,j}\alpha_0 - \omega_{i,j}), Enc_i(\omega_{i,j})\}_{1\leq j\leq m}$ to the semi-trusted third party C. Group member P_i sends $\{Enc_c(r'_{i,j})\}_{1\leq j\leq m}$ to C.

Upon receiving these values and decrypting them, third party C computes

$$\begin{aligned} z'_{i,j} =& \frac{z_{i,j}}{r'_{i,j}} + r_{i,j}\alpha_0 - \omega_{i,j} \\ =& r_{i,j}f(a_{i,j}) - \omega_{i,j}, \end{aligned} \tag{4}$$

and encrypts it with group member P_i's public key. Then C packages the intermediate result $Enc_i(z'_{i,j})$ and its corresponding $Enc_i(\omega_{i,j})$, and sends the packages $\{Enc_i(z'_{i,j}), Enc_i(\omega_{i,j})\}_{1\leq j\leq m}$ to each $P_i \in \mathcal{P}$ in random order (as illustrated in Fig. 3).

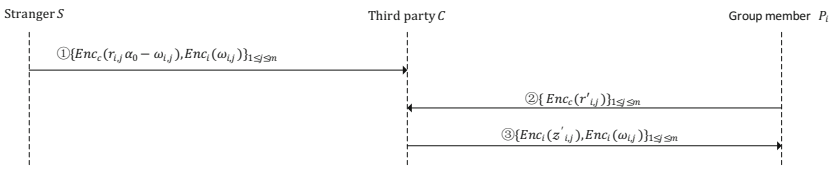

Fig. 3. When stranger S applies to join group $\mathcal{P}$, the *Application* phase will be executed

Group member $P(i)$ decrypts the received values with his private key and verifies

$$F'_{i,j} = z'_{i,j} + \omega_{i,j} \stackrel{?}{=} 0. \tag{5}$$

Because the values $f(a_{i,j})$ is randomized by random number $r_{i,j}$ generated by S in *Setup* phase, $P(i)$ will get zero or a random number from the result $F'_{i,j}$. If the equation is valid, then $a_{i,j}$ is a matched attribute. Otherwise it's an unmatched attribute. This is because if $a_{i,j}$ is a matched attribute, it is a root of polynomial $f(x)$, i.e., $f(a_{i,j}) = 0$. Since stranger S has sent $r_{i,j}\alpha_0$ to group member $P(i)$ in the first phase, S should utilize $\omega_{i,j}$ to re-randomize it in *Application* phase. Otherwise, $P(i)$ will learn which attribute the intermediate result $z'_{i,j}$ correspondences to.

After calculating all the results $F'_{i,j}$ and comparing them with zero, $P(i)$ will learn the size of the intersection set between him and stranger S. Then $P(i)$ can decide whether to agree to S's application.

5 Security Analysis

5.1 Security Under the HBC Model

Theorem 1. Assuming the semi-trusted third party C sends all the packages $\{Enc_s(z_{i,j}), Enc_s(r_{i,j}r'_{i,j}\alpha_0 + a_{i,j})\}_{1\le j\le m}$ to stranger S in random order and parameters $\{r'_{i,j}\}_{1\le j\le m}$ are random, we can achieve SG-1.

proof: In our scheme, sending all the packages in random order by C will blind from S the correspondence between P_i and the intermediate results $z_{i,j}$. In addition, $f(a_{i,j})$ and α_0 are randomized by $r_{i,j}r'_{i,j}$, so $F_{i,j}$ is an attribute in S's profile or a random number, and S can learn nothing from $r_{i,j}r'_{i,j}\alpha_0$. Thus stranger S just learns whether $F_{i,j}$ represents a matched attribute and what the matching attribute is, but can't learn its corresponding group member, any unmatched attributes or whether two computing results are from the same group member. Note that, to realize SG-1, $\{r_{i,j}\}_{1\le j\le m}$ don't have to be random.

Theorem 2. Assuming the semi-trusted third party C sends all the packages $\{Enc_i(z'_{i,j}), Enc_i(\omega_{i,j})\}_{1\le j\le m}$ to group member P_i in random order and parameters $\{r_{i,j}\}_{1\le j\le m}$ are random, we can achieve SG-2.

proof: In our protocol, if stranger S doesn't apply to join group $\mathcal{P}$, group members in $\mathcal{P}$ won't receive any responses from the semi-trusted third party C and they learn nothing about the matching results and S's profile.

If S applies to join group $\mathcal{P}$, since the packages $\{Enc_i(z'_{i,j}), Enc_i(\omega_{i,j})\}_{1\leq j\leq m}$ are sent to P_i by C in random order, and $r_{i,j}\alpha_0$ is re-randomized by $\omega_{i,j}$, P_i won't learn the corresponding attribute of the intermediate result $z'_{i,j}$. In addition, $\omega_{i,j}$ can't be equal to $r_{i,j}\alpha_0$ directly for P_i knowing the relationship between $r_{i,j}\alpha_0$ and its corresponding attribute. The value $f(a_{i,j})$ is randomized by $r_{i,j}$, so $F'_{i,j}$ is zero or a random number and group member P_i just learns weather $F'_{i,j}$ represents a matched attribute or not. Since $r_{i,j}$ is generated by stranger S for each attribute in group members' profiles, group members can learn nothing more than the matching results by exchanging information with each other. However, if the size of the intersection set between S and P_i equals to the size of P_i's own attribute set, i.e., $|\mathcal{A}_s \bigcap \mathcal{A}_i| = |\mathcal{A}_i|$, P_i will learn that all his attributes are in stranger S's profile. Note that, to realize SG-2, $\{r'_{i,j}\}_{1\leq j\leq m}$ don't have to be random.

Theorem 3. Assuming parameters $r\{_{i,j}\}_{1\leq j\leq m}$, $\{\tau_{i,k}\}_{1\leq k\leq n}$, $\{\omega_{i,j}\}_{1\leq j\leq m}$ are generated randomly, we can achieve SG-3.

proof: In any phase of our protocol, since the inputs received by the semi-trusted third party C are randomized, and some parameters used to calculate matching results are encrypted, C can learn nothing more than what can be derived from the values sent to him, his outputs and their corresponding group members. In the *Application* phase, even though C knows the values $r'_{i,j}$, it doesn't effect the security of our scheme.

5.2 Security Against Active Attacks

If Stranger S sets all coefficients $\{r_{i,j}\alpha_k\}_{0\leq k\leq n}$ of polynomial $r_{i,j}f(x)$ zero, the random numbers $r'_{i,j}$ in function $F_{i,j} = r_{i,j}r'_{i,j}f(a_{i,j}) + a_{i,j}$ won't work and then $F_{i,j}$ is equal to the attribute $a_{i,j}$. This kind of active attacks is referred to as zero polynomial attacks [3]. In our scheme, group member P_i sends $Enc_s(r_{i,j}r'_{i,j}\alpha_0 + a_{i,j})$ to stranger S, so merely setting $r_{i,j}\alpha_0$ zero, S can also realize zero polynomial attacks. To prevent this type of attacks, upon receiving the values $r_{i,j}\alpha_0$ from stranger S, P_i should first test $r_{i,j}\alpha_0 \stackrel{?}{=} 0$.

In order to increase the possibility of joining group $\mathcal{P}$, a malicious stranger S can use a large attribute set or launches the procedure many times to find out as many matched attributes in group members' profiles as possible. The former attack can be prevented by limiting the size of all users' attribute sets, the same approach as in [16]. The second attack can be prevented by auditing the times that stranger S runs the matching scheme to compute the intersection set with the same group in a short time by the semi-trusted third party C.

6 Performance Evaluation

In this section, we evaluate the performance of our scheme and compare it against the Gmatch scheme without batch verification [2]. We test the two schemes on the same hardware and OS, and our experimental environment is a 3.4GHz system with the OpenSSL library. We use RSA protocol to encrypt data to be transmitted and the length of the private/public key is 1024bits. In addition, we assume $|p| = 160$bits.

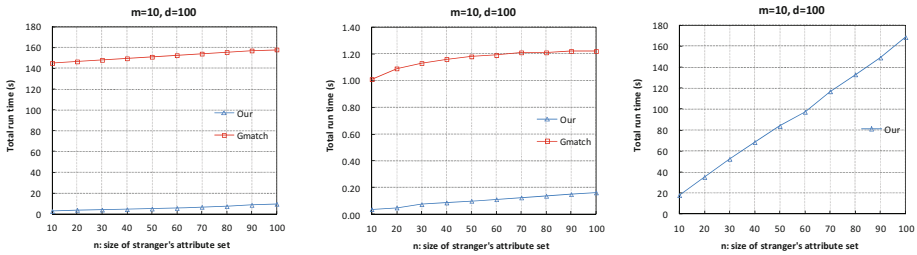

Fig. 4. Impact of n on total run time

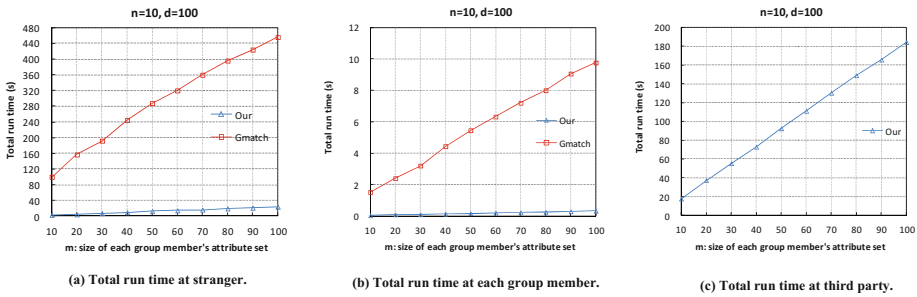

Fig. 5. Impact of m on total run time

In experiments, we change the size of stranger S's profile n, the size of group member P_i's profile m and the number of group members d respectively and measure the total run time at stranger S, group member P_i and the semi-trusted third party C. Since the Gmatch scheme doesn't include a third party, Fig. 4(c), Fig. 5(c) and Fig. 6(c) only show our system's run time on C. As shown in Fig. 4(a), Fig. 5(a) and Fig. 6(a), we can see that, our scheme is more efficient than the Gmatch scheme on S's client. Especially in Fig. 6(a), when only changing the number of group members d, the total run time of S increases linearly with d in our scheme, while the Gmatch scheme increases exponentially. From Fig.

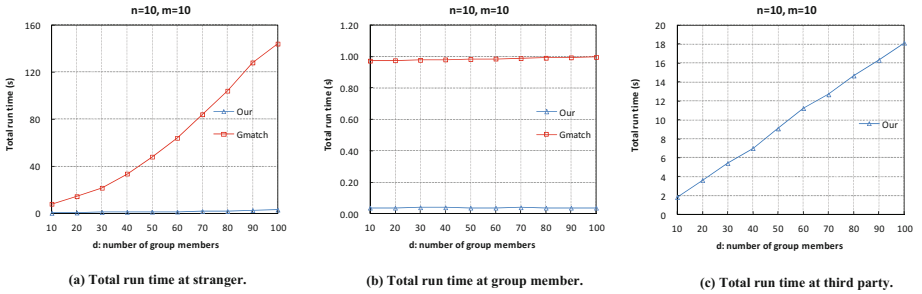

Fig. 6. Impact of d on total run time

4(b), Fig. 5(b) and Fig. 6(b), the run time at each group member is only linearly affected by n and m, while the Gmatch scheme is affected by n, m and d. This is because we don't use ring signature and additive homomorphic encryption in our scheme. Fig. 4(c), Fig. 5(c) and Fig. 6(c) show that the semi-trusted third party C's run time increase linearly with n, m and d. Although the total run time at C is much larger than that at stranger S and group member P_i, it is acceptable for users.

The performance evaluation of the two schemes show that in all settings our scheme is much more efficient and faster than the Gmatch scheme, and it is practical and lightweight enough in computation to be used on networked portable devices. That is because our scheme take advantage of the semi-trusted third party C to help calculate the polynomial and send the results to stranger S instead of using additive homomorphic encryption and ring signature. Our system relies mostly on modular multiplication while the Gmatch scheme included many exponentiation operations and bilinear operations. Considering the semi-trusted third party utilized in our system is easy to access and can be provided by service providers, the assumption of the existing of a third party is realizable in social networks.

7 Conclusion

In this paper, we propose a novel protocol to realize group matching by utilizing private set intersection (PSI) and a semi-trusted third party. During our scheme, by collecting different kinds of matching information, the stranger outside of the groups can make a better decision when choosing the most suitable group to join, and each group member can decide whether to agree to the stranger's application. We provide the thorough security analysis on our scheme and prove its security under honest-but-curious (HBC) model and against several certain active attacks. By comparison with an existing work, we show our system is practical and efficient in computation to be used in social networks.

Acknowledgement. This work was supported by National Natural Science Foundation of China under Grant 61232005, 61100237, 91118006.

References

1. Twitter, `http://twitter.com`
2. Wang, B., Li, B., Li, H.: Gmatch: Secure and privacy-preserving group matching in social networks. In: GLOBECOM, pp. 726–731 (2012)
3. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 1–19. Springer, Heidelberg (2004)
4. Dachman-Soled, D., Malkin, T., Raykova, M., Yung, M.: Efficient robust private set intersection. In: Abdalla, M., Pointcheval, D., Fouque, P.-A., Vergnaud, D. (eds.) ACNS 2009. LNCS, vol. 5536, pp. 125–142. Springer, Heidelberg (2009)
5. Hazay, C., Lindell, Y.: Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 155–175. Springer, Heidelberg (2008)
6. Jarecki, S., Liu, X.: Efficient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 577–594. Springer, Heidelberg (2009)
7. De Cristofaro, E., Tsudik, G.: Practical private set intersection protocols with linear complexity. In: Sion, R. (ed.) FC 2010. LNCS, vol. 6052, pp. 143–159. Springer, Heidelberg (2010)
8. Agrawal, R., Evfimievski, A., Srikant, R.: Information sharing across private databases. In: SIGMOD, pp. 86–97 (2003)
9. Vaidya, J., Clifton, C.: Secure set intersection cardinality with application to association rule mining. Journal of Computer Security 13(4), 593–622 (2005)
10. Kissner, L., Song, D.: Privacy-preserving set operations. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 241–257. Springer, Heidelberg (2005)
11. Camenisch, J., Zaverucha, G.M.: Private intersection of certified sets. In: Dingledine, R., Golle, P. (eds.) FC 2009. LNCS, vol. 5628, pp. 108–127. Springer, Heidelberg (2009)
12. Kim, M., Lee, H.T., Cheon, J.H.: Mutual private set intersection with linear complexity. In: Jung, S., Yung, M. (eds.) WISA 2011. LNCS, vol. 7115, pp. 219–231. Springer, Heidelberg (2012)
13. Dong, C., Chen, L., Camenisch, J., Russello, G.: Fair Private Set Intersection with a Semi-trusted Arbiter. IACR Cryptology ePrint Archive, p. 252 (2012)
14. Yang, Z., Zhang, B., Dai, J., Champion, A.C., Xuan, D., Li, D.: E-smalltalker: A distributed mobile system for social networking in physical proximity. In: IEEE ICDCS, pp. 468–477 (2010)
15. Lu, R., Lin, X., Liang, X., Shen, X.: A secure handshake scheme with symptoms-matching for mhealthcare social network. Mobile Networks and Applications 16(6), 683–694 (2011)
16. Li, M., Cao, N., Yu, S., Lou, W.: Findu: Privacy-preserving personal profile matching in mobile social networks. In: IEEE INFOCOM, pp. 2435–2443 (2011)