Asymmetric Social Proximity Based Private Matching Protocols for Online Social Networks

Arun Thapa, *Student Member, IEEE,* Ming Li, *Student Member, IEEE,*
Sergio Salinas, *Student Member, IEEE,* and Pan Li, *Member, IEEE*

Abstract—The explosive growth of Online Social Networks (OSNs) over the past few years has redefined the way people interact with existing friends and especially make new friends. Some works propose to let people become friends if they have similar profile attributes. However, profile matching involves an inherent privacy risk of exposing private profile information to strangers in the cyberspace. The existing solutions to the problem attempt to protect users' privacy by privately computing the private set intersection or private set intersection cardinality of the profile attribute sets of two users. These schemes have some limitations and can still reveal users' privacy. In this paper, we leverage community structures to redefine the OSN model and propose a realistic asymmetric social proximity measure between two users. Then, based on the proposed asymmetric social proximity, we design three private matching protocols, which provide different privacy levels and can protect users' privacy better than the previous works. We also analyze the computation and communication cost of these protocols. Finally, we validate our proposed asymmetric proximity measure using real social network data and conduct extensive simulations to evaluate the performance of the proposed protocols in terms of computation cost, communication cost, total running time, and energy consumption. The results show the efficacy of our proposed proximity measure and better performance of our protocols over the state-of-the-art protocols.

Index Terms—Online social networks; asymmetric social proximity; private matching protocols.

1 Introduction

Online Social Networks (OSNs) have had tremendous growth over the past few years. OSNs such as Facebook, Google$^+$, Linkedin are some of the most visited sites on the Internet [1], where users spend a significant fraction of their online time. Besides, increasing popularity of smart phones has extended the platforms used for accessing online social networks and provided a plethora of opportunities for mobile social networking. OSNs have redefined the way people interact with existing friends, and more importantly, make new friends. In particular, people can now explore potential friendships via OSNs, by looking for common interests, friends, and symptoms, close geographic proximity, etc., between each other. A naive solution to finding new friends in OSNs is using a server that stores all the users' information and conducting profile matching through the server. In this case, however, the server knows all the users' private information and becomes a single point of failure. Thus, if the server gets compromised, all users' privacy is at risk. For example, Twitter was attacked in early January 2013 and about 250,000 user accounts might have been compromised, with names and e-mails possibly being uncovered [2]. Facebook, Apple, Microsoft were under similar attacks in February 2013 [3]. Moreover, users may not have connectivity to the server all the time. Therefore, there has been growing interests in new privacy-preserving distributed solutions to finding friends in OSNs.

In OSNs and Mobile Social Networks (MSNs), many distributed solutions to privately finding the social proximity between two users have been proposed. The most common way of determining friendship between two people is through profile matching, i.e. finding out if they have common profile attributes, like interests [4], [5], symptoms [6]–[8], or some other social coordinates [9], [10]. In some cases, the number of common friends also serves as the proximity measure between two users [11], [12]. Such previous works employ various cryptographic tools to protect the privacy of the profile information of the users in the private matching process. After two strangers, say with profile attribute sets X and Y, execute a private matching protocol, the one who initiates the protocol will know either $X \cap Y$ or some function of $X \cap Y$ while the other one who responds does not know anything. Thus, a malicious user can execute the protocol with any user and leave without letting him/her do the same.

Moreover, most previous schemes for profile matching in online/mobile social networking are based on the premise that two people are likely to establish a social relationship only if they share similar profile attributes like interests, symptoms, or some other social coordinates. While it is true that people with similar profile attributes are likely to be friends, this is not the only way of determining friendship. For example, a doctor's best friend may not necessarily always be a doctor, but

- *This work was supported by the U.S. National Science Foundation under grants CNS-1149786 (CAREER Award), ECCS-1128768, and CNS-1147851.*
- *A. Thapa, M. Li, S. Salinas, and Pan Li are with the Department of Electrical and Computer Engineering, Mississippi State University, Mississippi State, MS 39762. E-mail: {at449@, ml845@, sas573@, li@ece.}msstate.edu.*

can be a writer who may share very few common profile attributes. In another example, two students who both have a lot of good friends studying in the Electrical and Computer Engineering department may become good friends, although they do not share many common profile attributes. We notice that whether two people can become friends not only depends on whether they have anything in common, but also is affected by whether their friends have anything in common. The intuition behind this is simple: a friend's friend can also be a friend.

In this paper, we leverage community structures to redefine the OSN model, and propose an asymmetric social proximity between two users. In particular, we consider that each OSN user is affiliated with some communities (or groups)[1], which the user weighs differently. We notice that the communities can actually tell a lot about their members. There can be a wide variety of communities in an OSN like a university community, a department community, a fan community of an artist, movies, or sports, and a community of certain professions. Besides, we notice that in real life people also value their friendships differently. Thus, we propose an asymmetric social proximity between two users, which is the cumulative weight of the common communities to one user considering both his/her and his/her friends' perceptions. We also design three different private matching protocols based on the proposed asymmetric social proximity. The main contributions of this paper can be briefly summarized as follows.

- We define an *asymmetric social proximity* measure between two users in an OSN, which considers both each user's and his/her friends' perceptions on the common communities between the two users. This proposed asymmetric social proximity can better capture the characteristics of making friends in OSNs.
- Based on the asymmetric social proximity, we design three different private matching protocols, i.e., L1P, L2P/EL2P, and L3P, which provide users with different privacy levels. In particular, our protocol L3P with the highest privacy level ensures that two users will not know any of their common communities before they become friends.
- We analyze the privacy, and computation and communication cost of the proposed protocols. Our protocols protect users' privacy better than the previous works based on symptoms, interests, and the number of common friends, with lower computation and communication cost. Particularly, in most previous schemes, e.g., [4]–[9], [11], a malicious user A can request friendship with another user B and then leave with B's private information before B knows anything about A. In our schemes, when one maliciousx user A requests friendship with another user B, A can know some limited private information of B's only if B is willing to accept the request.
- We validate our proposed asymmetric proximity measure using real social network data and conduct extensive simulations to evaluate the performance of the proposed protocols in terms of computation cost, communication cost, total running time, and energy consumption. The results show the advantages of our protocols over state-of-the-art protocols.

The rest of the paper is organized as follow. We discuss the related works in Section 2 and present our system model in Section 3. We detail the proposed three asymmetric social proximity based private matching protocols in Section 4. We present **simulation** results in Section 5, and finally conclude the paper in Section 6.

2 Related Work

In this section, we briefly introduce some previous studies that are most relevant to our work.

Private Set Intersection (PSI) protocols: In PSI protocols, two or more parties carrying their respective input sets interact to privately find the intersection set. In a two party (a server and a client) PSI protocol, the two parties interact so that the client learns only the intersection of the two input sets and the size of the server's input set, while the server learns nothing but the size of the client's input set. Since the introductory work of Freedman, Nissim, and Pinkas (FNP) [13], several PSI protocols [14]–[19] secure under semi-honest and/or malicious adversary models have been proposed. In such schemes, a client can artificially inflate its input set to learn the server's whole input set. Authorized PSI (APSI) protocols [20]–[22] avoid this problem by verifying the participants' inputs using some trusted authority. They involve expensive cryptographic processes, which lay heavy burdens on users' mobile devices.

Secure Multiparty Computation (SMC) protocols: SMC protocols allow two or more parties to privately calculate some functions of their inputs such that no party knows more than the function output and its own input. In particular, Yao [23] proposes the first SMC protocol based on garbled circuits. After that, there are a lot of works on improving security [24] and/or computation and communication complexities [25]–[30]. We do not employ SMC schemes for private proximity measurement in our scheme for two reasons. First, the generic SMC protocols are prohibitively expensive in both communication and computation. Second, our proposed social proximity measurement involves not only the users inputs (i.e., communities), but also their private parameters (i.e., α's and β's that will be introduced Section 3) which cannot be fed into the circuits to calculate proximity.

Social Proximity: The graph structure of social networks has been exploited to derive effective proximity measures. Katz measure [31] uses an ensemble of all the paths between two users in the network graph to

1. In what follows, we use "communities" and "groups" interchangeably.

derive the social proximity. Liben-Nowell et al. [32] and Tong et al. [33] also employ path-ensemble based methods for the future link prediction in social networks and proximity measurement. The path-ensemble based proximity measures are known to be effective in link prediction and proximity measurement in social networks as they capture more information about the underlying social network. However, they require the knowledge of the snapshot of the social network graph, and are prohibitively expensive in computation. Thus, these methods are not applicable to distributed proximity measurement.

Social Proximity Based Private Matching: Among distributed measurements of social proximity, one of the most common and simplest proximity measure is the number of common friends or profile attributes between two users of the network [4], [6]–[9], [11]. Intuitively, as the overlap between two user's profile attributes or friend spaces grows, their proximity increases. Based on distributed social proximity measurement, Zhang et al. [5] use homomorphic encryption to obtain fine-grained profile matching for mobile social networks. Similar profile matching schemes are presented in [4], [6], [9]–[11]. Profile matching in mobile health social network is studied in [7], [8] to privately match health profiles. Recently, Zhang et al. [34] proposes a mechanism to match-making profile search in a decentralized multi-hop mobile social network, where a user submits his/her "preference-profile" in order to search other users matching the profile. Similarly, Nagy et al. [12] presents a framework for finding common friends in a private manner using secure computation, set intersection, and bloom filters. Note that most of these studies focus on profile matching under the assumption that the social proximity between two users is symmetric, i.e., the social proximity calculated by each user is the same. In this paper, we utilize the communities and friend circles in an OSN to derive a realistic asymmetric social proximity in a distributed manner.

3 System Model

3.1 Network Model

Consider an online social network (OSN) where users store their own and friends' information on their devices. Such an OSN can be a decentralized OSN like that in [35], where no single server has information about all users, and two users can communicate via the Internet to establish a friendship. It can also be an MSN where two users' in close proximity can utilize bluetooth or WiFi to communicate for private matching. In addition, the network considered herein also includes the scenarios in centralized OSNs like Facebook, Google+, where users may not always be connected to the servers and can use the information stored in their mobile devices to find friends without the servers' involvement.

Note that we consider social friendships bidirectional, mutual, and reciprocating. In other words, if A is a friend of B's, B is also a friend of A's. Besides, we notice that in real life people value their friendships differently. Thus, as shown in Fig. 1, we propose that each user groups his/her friends into different friend circles like hometown friends, family friends, university friends, co-workers, and gym friends. In addition, we consider that each user i is affiliated with a set of communities, denoted by $C_i = \{C_i^1, C_i^2, ..., C_i^{c_i}\}$. The whole set of communities a user i or his/her friends are affiliated with, called "the overall community set" of user i and denoted by $\overline{C}_i$, is $\overline{C}_i = \bigcup_{j \in \overline{\mathcal{N}}_i} C_j$, where $\overline{\mathcal{N}}_i = \mathcal{N}_i \cup \{i\}$ and $\mathcal{N}_i$ denotes the set of user i's friends. We call a community in $\overline{C}_i$ one of user i's overall communities.

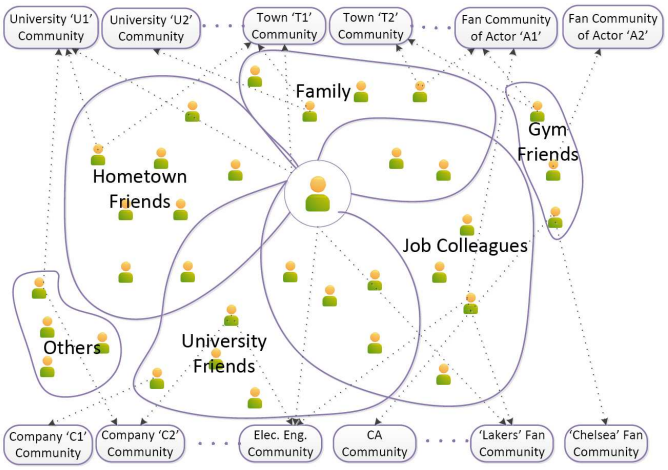


Fig. 1. System Model

3.2 Asymmetric Distributed Social Proximity Measurement

In order to measure the social proximity (denoted by Ψ) between two users in an OSN without revealing their privacy, we utilize the users' overall community sets instead of their private profiles. The intuition behind this is that two persons who both have a lot of close friends in the same several communities can probably be friends. In particular, we take the following parameters into account. First, as mentioned before, a user in an OSN divides his/her friends into different friend circles, which represent different friendship weights to the user. In particular, suppose a user i has a set of friend circles $FC_i = \{FC_i^1, FC_i^2, ..., FC_i^{f_i}\}$. In order to quantify the significance of a particular friend circle FC_i^j $(1 \leq j \leq f_i)$, user i assigns an integer value α_i^j $(0 \leq \alpha_i^j \leq \alpha^{max})$ to FC_i^j. A larger α_i^j indicates higher importance of the friend circle to the user. Second, each user, say i, also assigns an integer weight factor to each of the communities he/she is affiliated with, say C_i^j $(1 \leq j \leq c_i)$, which is denoted by $\beta_i(C_i^j)$ $(0 \leq \beta_i^j \leq \beta^{max})$. Note that α^{max} and β^{max} are predefined system parameters (integers) that are known to all the users.

Considering the above parameters, we define a community based social proximity between two users A and B as follows. Let $C_{AB} = \overline{C}_A \cap \overline{C}_B =$

$\{C^1_{AB}, C^2_{AB}, ..., C^{c_{AB}}_{AB}\}$, and $FC(i,j)$ denote a function which returns user i's friend circle(s) that i's friend, j, is in, i.e., $j \in FC^k_i$ for any $k \in FC(i,j)$. Besides, we define $FC(i,i) = \{0\}$ for any i, and $\alpha^0_j = \alpha^{max}$ for any j. Thus, the social proximity between A and B gauged by A is

$$\Psi_{A\leftarrow B} = \frac{\sum_{i=1}^{|C_{AB}|} \sum_{j \in C^i_{AB} \cap \overline{\mathcal{N}}_A} \left(\beta_j(C^i_{AB}) \sum_{\{k|k \in FC(A,j)\}} \alpha^k_A \right)}{\sum_{i=1}^{|\overline{C}_A|} \sum_{j \in C^i_A \cap \overline{\mathcal{N}}_A} \left(\beta_j(C^i_A) \sum_{\{k|k \in FC(A,j)\}} \alpha^k_A \right)}, \quad (1)$$

and that gauged by B is

$$\Psi_{B\leftarrow A} = \frac{\sum_{i=1}^{|C_{AB}|} \sum_{j \in C^i_{AB} \cap \overline{\mathcal{N}}_B} \left(\beta_j(C^i_{AB}) \sum_{\{k|k \in FC(B,j)\}} \alpha^k_B \right)}{\sum_{i=1}^{|\overline{C}_B|} \sum_{j \in C^i_B \cap \overline{\mathcal{N}}_B} \left(\beta_j(C^i_B) \sum_{\{k|k \in FC(B,j)\}} \alpha^k_B \right)}, \quad (2)$$

where $0 \leq \Psi_{A\leftarrow B} \leq 1$ and $0 \leq \Psi_{B\leftarrow A} \leq 1$. Evidently, the social proximity measures defined above are rational numbers. In (1), $\sum_{\{k|k \in FC(A,j)\}} \alpha^k_A$ is the total weight of friend j to A considering the multiple friend circles of A that j is in. $\beta_j(C^i_{AB})$ is the weight of one of the common communities, i.e., C^i_{AB}, shared by A and B to j. Thus, the numerator of the right-hand-side (RHS) of (1) represents the total equivalent weight of the common communities shared by A and B to A, considering both A's and A's friends' perceptions. Similarly, the denominator of the right-hand-side (RHS) of (1) represents the total equivalent weight of A's communities to A, considering both A's and A's friends' perceptions. Thus, (1) is the normalized weight of the common communities to A considering both A's and A's friends' perceptions. In other words, (1) quantifies how important the common communities are to A. Similarly, (2) is the normalized weight of the common communities to B considering both B's and B's friends' perceptions and quantifies how important the common communities are to B. Notice that when calculating $\Psi_{A\leftarrow B}$, A only needs his/her weights on his/her own friend circles and his/her friends' weights on As communities. In general, a larger $\Psi_{A\leftarrow B}$ indicates a closer social relationship of B to A.

Note that the proposed social proximity measurement is asymmetric, i.e., $\Psi_{A\leftarrow B}$ and $\Psi_{B\leftarrow A}$ are not necessarily equal. This is different from most of the distributed proximity measurements proposed for private matching, which are symmetric. We contend that asymmetric social proximity is more realistic, which is supported by a common intuition that the fact that A is the best friend of B does not necessarily mean B is the best friend of A.

3.3 Cryptographic Tools

3.3.1 Paillier Cryptosystem

Paillier designed an efficient asymmetric cryptosystem, called Paillier cryptosystem [36], based on decisional composite residuosity assumption. Due to its attractive additive homomorphic property, Paillier cryptosystem has been widely used in many applications like secure e-voting and private information retrieval. In particular, letting $\text{ENC}(\cdot)$ and $\text{DEC}(\cdot)$ denote the encryption and decryption functions of Paillier scheme, respectively, we have

- $\text{ENC}(m_1) \cdot \text{ENC}(m_2) = \text{ENC}(m_1 + m_2)$
- $\text{ENC}(m)^c = \text{ENC}(c \cdot m)$

The Paillier cryptosystem is semantically secure for sufficiently large public keys, which means that it is infeasible for a computationally bounded adversary to derive significant information about a message (plaintext) when given only its ciphertext and the corresponding public key. In this study, we assume that the public key is of 1184 bits for sufficient semantical security of the Paillier cryptosystem [5]. Therefore, a ciphertext is of 2048 bits, a Paillier encryption needs two 1024-bit exponentiations and one 2048-bit multiplication, and a Paillier decryption's cost is equivalent to one 2048-bit exponentiation.

Note that the proposed protocols can in fact work with any cryptosystem which is semantically secure and supports additive homomorphism. We employ Paillier cryptosystem to facilitate our illustrations in this paper.

3.3.2 The FNP Scheme

Freedman et al. [13] design a private matching and set intersection protocol, called FNP, using homomorphic encryption, on which our protocols are based. In FNP, a client constructs a polynomial $P(z) = (x_1 - z)(x_2 - z).....(x_{n_C} - z) = \sum_{k=0}^{n_C} u_k z^k$, where $x_1, x_2, ..., x_{n_C}$ are the elements of the client's input set X. The client then encrypts the coefficients and send $\text{ENC}(u_0), \text{ENC}(u_1), ..., \text{ENC}(u_{n_c})$ to the server. Utilizing the homomorphic property, the server constructs and evaluates the encrypted polynomial $\text{ENC}(P(z))$ at each of the element in its own input set Y. The server then chooses a random number ρ_i, and computes and returns to the client $\text{ENC}(\rho_i P(y_i) + y_i)$ for each $y_i \in Y$. When the client decrypts the ciphertext received from the server, it can find all $y_i \in X \cap Y$ as $P(y) = 0$ for all y_i's which are the roots of the polynomial $P(z)$ constructed by the client.

3.4 Adversary Model

Although there could be outsider adversaries trying to eavesdrop on the communications in the OSN, or modify, replay and inject messages, we focus on insider adversaries in our protocol design, who are the participators of the protocols and pose more challenges in protecting users' privacy. We believe, in the context of social networks, semi-honest or Honest But Curious

(HBC) adversary model best describes the characteristics of adversaries, which is considered as the adversary model in this study. A semi-honest adversary faithfully executes the protocols correctly but at the same time tries to gather more information about the other party than the protocols intend to disseminate.

4 Asymmetric Social Proximity Based Private Matching Protocols

In this section, we propose three novel and efficient social proximity based private matching protocols with different privacy levels. Before we delve into details, we first present some definitions below.

Definition 1: **Initiator**[2]: An Initiator is an OSN user who initiates a protocol for calculating social proximity. In other words, an Initiator is an OSN user who asks another user (a Responder) for friendship.

Definition 2: **Responder**: A Responder, upon the the request from an Initiator, replies by following the protocol.

Besides, when an Initiator asks a Responder for friendship, it should be the Responder who determines whether or not to accept the request by executing the protocol to find the social proximity.

4.1 Protocol for Level 1 Privacy (L1P)

The protocol ensuring level 1 privacy is suitable for users who decide to make friends with each other simply based on the common communities of their overall community sets. In this protocol, we first let the Responder learn the mutual communities and the size of the Initiator's input set ($\overline{C}_I$) (i.e., the Initiator's overall community set), while let the Initiator learn nothing but the size of the Responder's input set ($\overline{C}_R$) (i.e., the Responder's overall community set). Then, the Responder securely sends the common communities to the Initiator, if she confirms the request from the Initiator.

4.1.1 Protocol Details

We divide the protocol in two stages: offline and online. In order to speed up the matching process, the Initiator executes part of the protocol offline. In particular, the Initiator uses his input set $\overline{C}_I$ to construct the following polynomial:

$$P(z) = (\overline{C}_I^1 - z)(\overline{C}_I^2 - z).....(\overline{C}_I^{|\overline{C}_I|} - z) = \sum_{k=0}^{|\overline{C}_I|} u_k z^k \quad (3)$$

where $\overline{C}_I^i \in \overline{C}_I$ $(1 \leq i \leq |\overline{C}_I|)$. He then encrypts the coefficients u_k's of the polynomial and obtains $\mathrm{ENC_I}(u_0), \mathrm{ENC_I}(u_1), ..., \mathrm{ENC_I}(u_{|\overline{C}_I|})$, where $\mathrm{ENC_I}(\cdot)$ is the Initiator's homomorphic encryption function.

As shown in Fig. 1 in Appendix A (available in the online supplemental material), in the online stage, the Initiator first sends the encrypted coefficients along with his public key to the Responder. The Responder subsequently constructs the encrypted polynomial based on the encrypted coefficients utilizing the homomorphic property, i.e.,

$$\begin{aligned}\mathrm{ENC_I}(P(z)) \quad = \quad & \mathrm{ENC_I}(u_0)^{z^0} \cdot \mathrm{ENC_I}(u_1)^{z^1} \cdot \\ & .... \cdot \mathrm{ENC_I}(u_{|\overline{C}_I|})^{z^{|\overline{C}_I|}}. \end{aligned} \quad (4)$$

The Responder then evaluates $\mathrm{ENC_I}(P(z))$ at each of her own input element, computes the following function, and sends it along with her public key to the Initiator:

$$\mathrm{ENC_I}(P(\overline{C}_R^i) + R_i) = \mathrm{ENC_I}(P(\overline{C}_R^i)) \cdot \mathrm{ENC_I}(R_i)$$

where R_i is a random ID generated by the Responder for the community corresponding to $\overline{C}_R^i$, and of the same length as $P(\overline{C}_R^i)$. Then, in the second step, the two parties engage in a challenge response protocol to establish a shared secret key. In particular, the Initiator chooses a random nonce K as the key for a predefined symmetric encryption function $E(\cdot)$ (e.g., AES), encrypts it with the Responder's public key, and sends $ENC_R(K)$ to the Responder, where $ENC_R(\cdot)$ is the Responder's homomorphic encryption function. The Responder recovers K and acknowledges with $ENC_I(K+1)$ to the Initiator. Both parties use K as the shared secret key in the third step. Finally, in the third step, the Initiator decrypts the data received from the Responder in the first step, i.e., $\mathrm{ENC_I}(P(\overline{C}_R^i) + R_i)$'s, encrypts the decrypted data with the symmetric key K using the symmetric encryption algorithm $E_K(\cdot)$, and then sends $E_K\big(DEC_I(ENC_I(\overline{C}_R^i) + R_i))\big) = E_K\big(P(\overline{C}_R^i) + R_i\big)$ back to the Responder. Note that $P(\overline{C}_R^i) = 0$ when the corresponding community $\overline{C}_R^i$ is a mutual community between the Initiator's and the Responder's overall community sets. Thus, after recovering $P(\overline{C}_R^i) + R_i$, the Responder can know the mutual communities by checking R_i's. If she does not want to make friends with the Initiator, she can either ignore or decline the request. Otherwise, she encrypts the mutual communities with the shared secret K and sends $E_K(\overline{C}_I \cap \overline{C}_R)$ to the Initiator, who can now find the shared communities. If he would like to continue, he can finally become friends with the Responder. Note that to prevent some Initiators from possibly knowing some of the Responder's communities by colluding with each other, the Responder generates a new R_i corresponding to $\overline{C}_R^i$ upon each friendship request.

4.1.2 Protocol Analysis

In the following, we analyze the privacy of, and the communication cost and computation cost of the protocol.

Privacy Analysis: Here we analyze the privacy of the protocol.

***Theorem** 1: Before they become friends, the Initiator only learns $|\overline{C}_R|$, and $\overline{C}_I \cap \overline{C}_R$ if the Responder confirms his request, while the Responder only learns $|\overline{C}_I|$ and $\overline{C}_I \cap \overline{C}_R$.*

2. Without loss of generality, we use masculine pronouns for an Initiator and feminine pronouns for a Responder.

Proof: The Initiator uses semantically secure homomorphic encryption to encrypt the coefficients of the polynomial P, whose roots are the elements of his input set $\overline{C}_I$. The Responder cannot decrypt or distinguish the coefficients, and hence cannot know $\overline{C}_I$ but can learn $|\overline{C}_I|$. Following the protocol, the Responder then sends $\mathrm{ENC_I}(P(\overline{C}_R^i)+R_i)$'s back to the Initiator, where R_i's are random numbers of the same length as $P(\overline{C}_R^i)$'s. Thus, the Initiator can only learn $|\overline{C}_R|$ but nothing more. After receiving $P(\overline{C}_R^i)+R_i$ from the Initiator, the Responder will be able to figure out $\overline{C}_I \cap \overline{C}_R$, and let the Initiator know as well if she decides to confirm the request. Otherwise, the protocol terminates and both parties do not know anything further about each other. □

Computation and Communication Costs: The total computation cost and communication cost in this protocol can be analyzed similar to those in [13]. Differently, in the proposed L1P, the Initiator executes part of the protocol offline which in turn reduces the online computation cost. Specifically, the Initiator, in the offline stage, computes the polynomial $P(z)$ and encrypts its coefficients with his public key. As the computational complexity of the exponentiation operation dominates the other operations like multiplication and addition, we analyze the computation overhead focusing on exponentiation operations. Recall that the input set size of the Initiator and of the Responder are $|\overline{C}_I|$ and $|\overline{C}_R|$ respectively, the offline computation cost of the Initiator is $O(|\overline{C}_I|)$ exponentiations. In the online stage, the Initiator's computation cost is $O(|\overline{C}_R|)$ due to decrypting $\mathrm{ENC_I}(P(\overline{C}_R^i)+R_i)$'s received from the Responder. The Responder's computation cost for constructing the encrypted polynomial and evaluating at each of her inputs is $O(|\overline{C}_R|\log\log\ |\overline{C}_I|)$ exponentiations, considering that the polynomial can be efficiently evaluated by Hornor's rule and the balanced bucket allocation scheme presented in [13].

Regarding the communication cost, the Initiator first transmits $O(|\overline{C}_I|)$ encrypted coefficients and the Responder returns $O(|\overline{C}_R|)$ ciphertexts to the Initiator. Subsequently, in the next step, the Initiator returns $O(|\overline{C}_R|)$ decrypted messages and the Responder returns $O(|\overline{C}_I \cap \overline{C}_R|)$ common communities. Thus, the total communication cost for the Initiator is $O(|\overline{C}_I|+|\overline{C}_R|)$ and that for the Responder is $O(|\overline{C}_R|+|\overline{C}_I \cap \overline{C}_R|)$.

Moreover, the L1P protocol allows parallel processing in communication and computation which can further reduce the online execution time. In particular, the Responder does not have to wait for all the coefficients before beginning the computation of the encrypted polynomial. Similarly, when she starts returning the evaluated encrypted polynomial at each of her input, the Initiator can start decrypting the ciphertexts as soon as he receives one. Hence, if the communication cost is equal to or greater than the online computation overhead in time, the total communication cost would approximately be the total execution time of the protocol.

4.2 Protocol for Level 2 Privacy (L2P)

In the protocol for level 1 privacy (L1P), the Responder determines whether or not to accept the Initiator's request for a social friendship only based on their common overall communities, which may not characterize the social proximity well. In this section, we design a protocol for level 2 privacy, called L2P, utilizing the proposed community based asymmetric social proximity measurement. This protocol is suitable for the case when the Initiator is willing to establish a friendship relation with the Responder but the Responder accepts the relationship only if her requirement on the friendship is fulfilled. In particular, in L2P, the Responder accepts the friendship request from the Initiator if the social proximity measured by her, i.e, $\Psi_{R\leftarrow I}$, is greater than a threshold predefined by herself, denoted by Ψ_{R_τ}. The protocol is detailed as follows.

4.2.1 Protocol Details

Similar to that in L1P, an Initiator and a Responder can execute part of the protocol offline in order to speed up the matching process.

OFFLINE: The same as that in L1P, the Initiator constructs the polynomial P, with his inputs $\overline{C}_I = \{\overline{C}_I^1, \overline{C}_I^2, ...\overline{C}_I^{|\overline{C}_I|}\}$ being the roots, and encrypts the coefficients using his own homomorphic encryption function $\mathrm{ENC_I}(\cdot)$. On the other hand, the Responder calculates the partial social proximity corresponding to each of her overall communities as follows:

$$\Psi_{R\leftarrow I}^i = \frac{\sum_{j\in\overline{C}_R^i\cap\overline{\mathcal{N}}_R}\left(\beta_j(\overline{C}_R^i)\sum_{\{k|k\in FC(R,j)\}}\alpha_R^k\right)}{\sum_{i=1}^{|\overline{C}_R|}\sum_{j\in\overline{C}_R^i\cap\overline{\mathcal{N}}_R}\left(\beta_j(\overline{C}_R^i)\sum_{\{k|k\in FC(R,j)\}}\alpha_R^k\right)} \quad (5)$$

for any $\overline{C}_R^i \in \overline{C}_R$ $(1\le i\le|\overline{C}_R|)$. The Responder needs to encrypt the partial social proximity for all $\overline{C}_R^i$'s with her public key. However, $\Psi_{R\leftarrow I}^i$ is a fractional number and general additive homomorphic schemes cannot be used to encrypt the fractional numbers. Note that since α's and β's are integers, $\Psi_{R\leftarrow I}^i$ is a rational number. Besides, the denominator in (5) is a constant for all $\overline{C}_R^i \in \overline{C}_R$. We denote the denominator by D_R. The Responder encrypts the numerator (integer) of the partial social proximity with her public key, i.e., computes $\mathrm{ENC_R}(\Psi_{R\leftarrow I}^i \cdot D_R)$, where $\mathrm{ENC_R}(\cdot)$ is her homomorphic encryption function. In addition, the Responder assigns a random ID, R_i, to each of her overall communities upon each friendship request.

ONLINE: When an Initiator and a Responder decide to execute the protocol, the Initiator first sends the encrypted coefficients of the polynomial P to the Responder. Note that the Initiator and the Responder

exchange their public keys to establish a shared secret key K in the same way as that in L1P. The detailed description of shared key establishment is omitted below to avoid redundancy. The Responder then constructs the encrypted polynomial according to (4), and evaluates the polynomial at each of her input $\overline{C}_R^i \in \overline{C}_R$. Taking advantage of the homomorphic property of the encryption, the Responder further constructs the following message

$$(A_i, B_i, C_i) = \Big(\mathrm{ENC_I}(\rho_i \cdot P(\overline{C}_R^i)), \quad \mathrm{ENC_I}\left(P(\overline{C}_R^i) + \mathrm{ENC_R}(\Psi_{R\leftarrow I}^i \cdot D_R)\right), R_i \Big) \tag{6}$$

for each $\overline{C}_R^i \in \overline{C}_R$, where ρ_i is a random number of the same length as $P(y_i)$, and sends (A_i, B_i, C_i) to the Initiator.

The Initiator then decrypts A_i, and for each i with $\mathrm{DEC_I}(A_i) = 0$, calculates $\mathrm{DEC_I}(B_i) = (0 + \mathrm{ENC_R}(\Psi_{R\leftarrow I}^i) \cdot D_R) = \mathrm{ENC_R}(\Psi_{R\leftarrow I}^i \cdot D_R)$, which implies the corresponding input $\overline{C}_R^i \in (\overline{C}_I \cap \overline{C}_R)$. After that, the Initiator can calculate the encrypted social proximity for the Responder by aggregating all B_i's as follows:

$$\begin{aligned}
&\mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D) \\
&= \prod_{\{i|\mathrm{DEC_I}(A_i)=0\}} \mathrm{DEC_I}(B_i) \\
&= \prod_{\{i|\mathrm{DEC_I}(A_i)=0\}} \mathrm{ENC_R}\Bigg(\sum_{j\in \overline{C}_R^i \cap \overline{\mathcal{N}}_R} \Big(\beta_j(\overline{C}_R^i) \cdot \sum_{\{k|k\in FC(R,j)\}} \alpha_R^k\Big)\Bigg) \\
&= \mathrm{ENC_R}\Bigg(\sum_{i=1}^{|C_{IR}|} \sum_{j\in \overline{C}_R^i \cap \overline{\mathcal{N}}_R} \Big(\beta_j(\overline{C}_R^i) \sum_{\{k|k\in FC(R,j)\}} \alpha_R^k\Big)\Bigg) \\
&= \mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)
\end{aligned} \tag{7}$$

Obviously, we can see that when $A_i = 0$, the term B_i gives the numerator of the encrypted (by the Responder) partial social proximity attributed to the community C_R^i that is common to both the Initiator and the Responder. Thus, due to homomorphic property, the product of the encryption over all the communities with $A_i = 0$ is equal to the encryption of the sum of the partial social proximities attributed to all the common communities shared by the Initiator and the Responder, as shown in (7). As is evident from (2), (7) is in fact the encrypted (by the Responder) social proximity between the Responder and the Initiator gauged by the Responder times D_R, i.e., $ENC_R(\Psi_{R\leftarrow I} \cdot D_R)$.

The Initiator sends $\mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)$ to the Responder, who decrypts it and checks to see if $\Psi_{R\leftarrow I} \cdot D_R \geq \Psi_{R_\tau} \cdot D_R$, i.e. $\Psi_{R\leftarrow I} \geq \Psi_{R_\tau}$. If not, the Responder aborts the protocol and informs the Initiator. Otherwise, the Responder confirms the Initiator's request, who then encrypts the R_i's, corresponding to the cases when $\mathrm{DEC_I}(A_i) = 0$, with the symmetric key K and sends $E_K(R_i)$'s to the Responder. After decrypting $E_K(R_i)$, the Responder can then know $\overline{C}_I \cap \overline{C}_R$ and sends $E_K(\overline{C}_I \cap \overline{C}_R)$ back to the Initiator, who can now become friends with the Responder if he still would like to proceed.

4.2.2 Extended Protocol for Level 2 Privacy (EL2P)

In the above L2P protocol, it is possible that the Responder may learn more than just the social proximity $\Psi_{R\leftarrow I}$ when receiving $\Psi_{R\leftarrow I} \cdot D_R$ and hence $\Psi_{R\leftarrow I}$ (the Responder knows D_R) from the Initiator. For example, if there happens to be only one common community between the Initiator and the Responder, then it is possible for the Responder to find out the common community by looking at the partial social proximity $\Psi_{R\leftarrow I}^i$ value of each of her communities even if $\Psi_{R\leftarrow I} \not\geq \Psi_{R_\tau}$. Similarly, even if there are multiple common communities shared by the Initiator and the Responder, the Responder may learn the common communities by checking if the sum of several partial social proximity is equal to $\Psi_{R\leftarrow I}$ received from the Initiator. Here, we extend the L2P protocol so that the Responder only learns whether $\Psi_{R\leftarrow I} \geq \Psi_{R_\tau} + \epsilon/D_R$, where ϵ is a small number such that $\epsilon \ll \Psi_{R_\tau}$, instead of the value of $\Psi_{R\leftarrow I}$. The detailed process of EL2P is described in Fig. 2, in Appendix A, available in the online supplemental material.

Specifically, at the end of step 1) of the online phase, the Responder sends $\mathrm{ENC_R}(\Psi_{R_\tau} \cdot D_R)$ in addition to (A_i, B_i, C_i) to the Initiator. The Initiator then computes $\mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)$ according to (7), and chooses three large positive random numbers r_1, r_2, and r_3 such that $0 \ll r_1 < r_2 < r_3$ and $\epsilon < \frac{r_2 - r_1}{r_3} \ll \Psi^{min}$, where Ψ^{min} is a predefined minimum social proximity threshold and known to all the users. Note that

$$\begin{aligned}
& (r_1 + r_3(\Psi_{R\leftarrow I} \cdot D_R)) \geq (r_3(\Psi_{R_\tau} \cdot D_R) + r_2) \\
\Longrightarrow\ & \Psi_{R\leftarrow I} \geq \left(\Psi_{R_\tau} + \frac{(r_2 - r_1)/r_3}{D_R}\right) \\
\Longrightarrow\ & \Psi_{R\leftarrow I} \geq \Psi_{R_\tau} + \epsilon/D_R
\end{aligned}$$

Therefore, the Initiator can compute $(M, N) = \big(\mathrm{ENC_R}(r_1 + r_3(\Psi_{R\leftarrow I} \cdot D_R)), \mathrm{ENC_R}(r_3(\Psi_{R_\tau} \cdot D_R) + r_2)\big)$ as follows

$$\begin{aligned}
& \mathrm{ENC_R}(r_1 + r_3(\Psi_{R\leftarrow I} \cdot D_R)) \\
=\ & \mathrm{ENC_R}(r_1) \cdot \mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)^{r_3} \\
& \mathrm{ENC_R}(r_3(\Psi_{R_\tau} \cdot D_R) + r_2) \\
=\ & \mathrm{ENC_R}(\Psi_{R_\tau} \cdot D_R)^{r_3} \cdot \mathrm{ENC_R}(r_2)
\end{aligned}$$

and sends (M, N) back to the Responder (instead of sending $\mathrm{ENC_R}(\Psi_{R\leftarrow I} \cdot D_R)$ to her). The Responder then checks to see if $\mathrm{DEC_R}(M) \geq \mathrm{DEC_R}(N)$ and follows the rest of the protocol accordingly in the same way as presented above.

We can see that in this extended protocol EL2P, the Responder is only able to learn if $\Psi_{R\leftarrow I} \geq \Psi_{R_\tau} + \epsilon/D_R$, i.e., $\Psi_{R\leftarrow I} > \Psi_{R_\tau}$ (since $\epsilon/D_R \ll \Psi^{min}/D_R \ll \Psi_{R_\tau}$), and the above problem can be addressed.

4.2.3 Protocol Analysis

In the following we analyze the EL2P protocol in terms of privacy, and computation and communication cost.

Privacy Analysis: We first analyze the privacy of the protocol EL2P.

***Theorem** 2: Before they become friends, the Initiator learns only $|\overline{C}_R|$ and $|\overline{C}_I \cap \overline{C}_R|$, and the mutual communities $\overline{C}_I \cap \overline{C}_R$ if $\Psi_{R\leftarrow I} > \Psi_{R_\tau}$, while the Responder learns only $|\overline{C}_I|$, and the mutual communities $\overline{C}_I \cap \overline{C}_R$ if $\Psi_{R\leftarrow I} > \Psi_{R_\tau}$.*

Proof: The Initiator uses semantically secure homomorphic encryption to encrypt the coefficients of the polynomial P, whose roots are the elements of his input set $\overline{C}_I$. The Responder cannot decrypt or distinguish the coefficients, and hence cannot know $\overline{C}_I$ but can learn $|\overline{C}_I|$. Following the protocol, the Responder then sends (A_i, B_i, C_i)'s to the Initiator, who can then know $|\overline{C}_R|$. By decrypting (A_i, B_i, C_i)'s and counting all A_i's that are decrypted to be 0, the Initiator can then know the size of the mutual community set, i.e., $|\overline{C}_I \cap \overline{C}_R|$, but does not know which the mutual communities are. He then computes the tuple (M, N), and sends it to the Responder. If the Responder finds $\Psi_{R\leftarrow I} > \Psi_{R_\tau}$, she informs the Initiator who sends her the random IDs R_i's, and hence can know the mutual communities $\overline{C}_I \cap \overline{C}_R$. Otherwise, the protocol terminates and both parties do not know anything further about each other. Besides, similarly to that in Theorem 1, the Initiator and the Responder cannot know all the communities in each other's overall community set by artificially extending their input sets.

Moreover, one may argue that it is possible for the Initiator to cheat by increasing M, for example, computing $M' = M \cdot \mathrm{ENC_R}(r_4)$ where $r_4 > 0$ or $M' = M^{r_4}$ where $r_4 > 1$, so that the Responder will get $\mathrm{DEC_R}(M) > \mathrm{DEC_R}(N)$ and hence accepts his request. However, the Initiator will always be caught since the Responder can verify in step 3) of the online phase whether or not $\Psi_{R\leftarrow I} > \Psi_{R_\tau}$ by checking the received R_i's from the Initiator before revealing $\overline{C}_I \cap \overline{C}_R$ to the Initiator. Without receiving the mutual communities, the Initiator cannot finally be authorized to make friends with the Responder. □

Computation and Communication Costs: The Initiator and the Responder execute part of the protocol offline, as in L1P, which can reduce the online computation time. In particular, in the offline phase, the Initiator incurs $O(|\overline{C}_I|)$ exponentiations to compute the encrypted coefficients of $P(z)$. Similarly, the Responder has a computation load of $O(|\overline{C}_R|)$ exponentiations to compute the partial social proximity offline. In the online phase, the computation cost for the Initiator is $O(|\overline{C}_R|)$ exponentiations (in step 2) of the online phase as shown in Fig. 2 in Appendix A). The Responder performs $O(|\overline{C}_R| \log\log |\overline{C}_I|)$ exponentiations in step 1) of the online phase.

As for the communication cost, the Initiator sends $O(|\overline{C}_I|)$ encrypted coefficients in step 1) and $O(|\overline{C}_I \cap \overline{C}_R|)$ IDs (R_i's) in step 3) of the protocol. The Responder, on the other hand, replies with $O(|\overline{C}_R|)$ encrypted ciphertexts in step 1) and $O(|\overline{C}_I \cap \overline{C}_R|)$ communities in step 3). Thus, the total communication overhead for the Initiator is $O(|\overline{C}_I| + |\overline{C}_I \cap \overline{C}_R|)$ and that for the Responder is $O(|\overline{C}_R| + |\overline{C}_I \cap \overline{C}_R|)$.

4.3 Protocol for Level 3 Privacy

In the L2P protocol, the Responder determines whether or not to be friends with the Initiator based on the community based social proximity, while the Initiator still can only make his final decision based on their common communities. Besides, in terms of privacy, in L2P the Responder will know $\overline{C}_I \cap \overline{C}_R$ if $\Psi_{R\leftarrow I} > \Psi_{R_\tau}$, no matter whether the social proximity measured by the Initiator is large enough or not. In this section, we develop a protocol for level 3 privacy, called L3P, to address the above problems. This protocol is suitable for users with very high privacy requirements. In this protocol, both the Initiator and the Responder make sure their requirements on friendship are fulfilled before revealing any matching information to each other. If either of the requirements is not satisfied, neither of them knows the matching profile information, i.e., the common communities $\overline{C}_I \cap \overline{C}_R$.

4.3.1 Protocol Description

The same as that in L1P and L2P, part of the L3P protocol can be completed offline. In what follows, we briefly describe the offline and online phases of the protocol, respectively, which are also shown in Fig. 3 in Appendix A, available in the online supplemental material.

OFFLINE: In the offline phase, the Initiator constructs a polynomial P with his input set $\overline{C}_I$ being the roots, while the Responder constructs a polynomial Q with her input set $\overline{C}_R$ being the roots (step 1)). Each of them encrypts the coefficients of their polynomials using their own public keys in step 2), and computes partial community based social proximities in step 3).

ONLINE: In online phase, the Initiator and the Responder exchange their encrypted coefficients in step 1). The Initiator and the Responder construct $\mathrm{ENC_R}(Q(z))$ and $\mathrm{ENC_I}(P(z))$, respectively, based on the received ciphertexts, and evaluate at each of their own inputs, and exchange their tuples (A'_i, B'_i) and (A_i, B_i), i.e.,

$$\big(\mathrm{ENC_R}(\rho'_i \cdot P(\overline{C}^i_I)), \mathrm{ENC_R}(\rho'_i \cdot P(\overline{C}^i_I) + \mathrm{ENC_I}(\Psi^i_{I\leftarrow R} \cdot D_I)\big)$$
$$\big(\mathrm{ENC_I}(\rho_i \cdot P(\overline{C}^i_R)), \mathrm{ENC_I}(\rho_i \cdot P(\overline{C}^i_R) + \mathrm{ENC_R}(\Psi^i_{R\leftarrow I} \cdot D_R)\big)$$

along with $\mathrm{ENC_I}(\Psi_{I_\tau} \cdot D_I)$ and $\mathrm{ENC_R}(\Psi_{R_\tau} \cdot D_R)$, respectively, in step 2). Note that ρ'_i and ρ_i are random numbers of the same length as $P(\cdot)$, Similar to that in step 2) of the L2P online phase, the Initiator and the Responder exchange the tuples (M', N') and (M, N) in step 3), i.e.,

$$(\mathrm{ENC_R}(r'_1 + r'_3(\Psi_{R\leftarrow I} \cdot D_R)), \mathrm{ENC_R}(r'_3(\Psi_{R_\tau} \cdot D_R) + r'_1))$$
$$(\mathrm{ENC_I}(r_1 + r_3(\Psi_{I\leftarrow R} \cdot D_I)), \mathrm{ENC_I}(r_3(\Psi_{I_\tau} \cdot D_I) + r_1)).$$

where $0 \ll r'_1 < r'_2 < r'_3$ and $\epsilon < \frac{r'_2 - r'_1}{r'_3} \ll \Psi^{min}$, and $0 \ll r_1 < r_2 < r_3$ and $\epsilon < \frac{r_2 - r_1}{r_3} \ll \Psi^{min}$. If at least one of the social proximity criteria is not satisfied, i.e., if $\Psi_{I \leftarrow R} \not> \Psi_{I_\tau}$ or/and $\Psi_{R \leftarrow I} \not> \Psi_{R_\tau}$, they cannot become friends and the protocol stops at step 4) before either of them is able to learn any matching information. Otherwise, i.e., if $\Psi_{I \leftarrow R} > \Psi_{I_\tau}$ and $\Psi_{R \leftarrow I} > \Psi_{R_\tau}$ both hold, the Initiator and the Responder are both assumed to be willing to establish a social friendship and they can become friends now.

4.3.2 Protocol Analysis

Next, we present the analysis on the privacy, and computation and communication cost of the L3P protocol.

Privacy Analysis: The privacy of the L3P protocol is analyzed as follows.

Theorem *3: Before they become friends, the Initiator learns* $|\overline{C}_R|$ *and* $|\overline{C}_I \cap \overline{C}_R|$*, while the Responder learns* $|\overline{C}_I|$ *and* $|\overline{C}_I \cap \overline{C}_R|$.

Proof: The proof is similar to that of Theorem 2 and hence omitted here. □

Moreover, in most previous schemes, e.g., [4]–[9], [11], a user can request friendship with another user, run some protocols, and then leave with the user's private information before the user can know anything. In our schemes, as shown in Theorems 1-3, when one user requests friendship with another, he/she can know some of the user's important private information only if the user is willing to accept the request.

Computation and Communication Costs: A significant fraction of the computation in L3P can be done offline. In particular, as shown in Fig. 3 in Appendix A, the Initiator performs $(|\overline{C}_I|+1)$ encryptions on the coefficients of the polynomial $P(z)$ and also computes $|\overline{C}_I|$ encryptions on partial social proximity measurements. Thus, the Initiator's total offline computation complexity is $O(|\overline{C}_I|)$ exponentiations. Similarly, the Responder's total offline computation complexity is $O(|\overline{C}_R|)$ exponentiations. In the online phase, following the similar analysis to that of the previous two protocols, the Initiator's computation complexity in step 2) is $O(|\overline{C}_I| \log\log |\overline{C}_R|)$ exponentiations and that in step 3) is $O(|\overline{C}_R|)$ exponentiations. The Responder's computation complexity is $O(|\overline{C}_R| \log\log |\overline{C}_I|)$ and $O(|\overline{C}_I|)$ exponentiations in step 2) and step 3), respectively.

Moreover, the Initiator's communication cost is $O(|\overline{C}_I|)$ and $O(|\overline{C}_R|)$ in step 1) and step 2), respectively. Similarly, the Responder's communication cost is $O(|\overline{C}_R|)$ and $O(|\overline{C}_I|)$ in step 1) and step 2), respectively. Therefore, both the Initiator and the Responder have a total communication cost of $O(|\overline{C}_R| + |\overline{C}_I|)$. In addition, as mentioned before, some computation and communication can be done in parallel, thus reducing the overall protocol execution time.

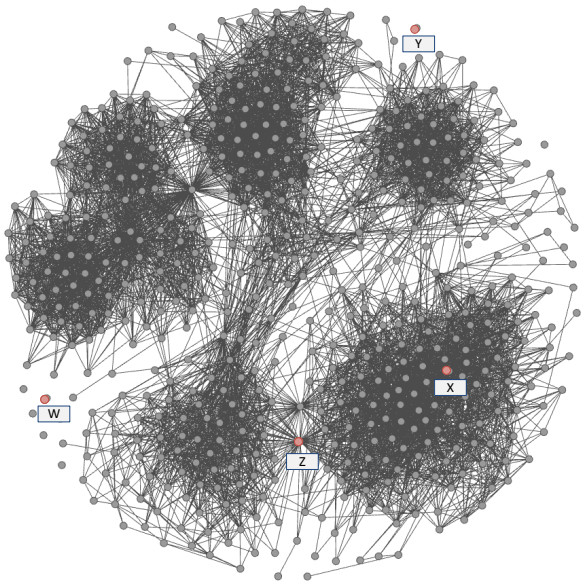


Fig. 2. Facebook Ego-network of One Author A.

5 Performance Evaluation

5.1 Asymmetric Social Proximity Measure Validation

In Section 3.2, we propose an asymmetric social proximity metric between two users, which is based on each user's as well as his/her friends' perceptions on the common communities between the two users. In this section, we design an experiment to validate the proposed metric using one author's (whom we denote by A) Facebook ego-network as shown in Fig. 2. The ego-network has 556 nodes (A's friends) and 7856 edges (interconnections among A's friends). The degree of each node in the network gives the number of common friends between A and the node (A's friend). Note that A is not in the network.

In order to quantify the proximity between A and any of his friends according to the asymmetric proximity metric proposed in this paper, A divides his friends into the following six friend circles: $FC_A = \{FC_A^1, FC_A^2, FC_A^3, FC_A^4, FC_A^5, FC_A^6\}$ = {Friends from hometown, Friends in the current university, Friends from the previous university, Job 1 friends, Job 2 friends, Others}[3]. Starting clockwise from the large cluster in the lower right in Fig. 2, the clusters correspond to FC_A^1 to FC_A^5 respectively. We look at each node (A's friend) in the network, and associate it with one or more communities according to its current and previous locations, occupations, academic institutions, etc. For example, a node V in the network can be a member of city T_i and

3. Interestingly, the different clusters in the ego-network as shown in Fig. 2 approximately represent these different friend circles of A (except "Others"). This opens up the possibility of automating the process of dividing one's friends into different friend circles. The weights on the friend circles can be estimated automatically, e.g., based on the number of friends in them, and finally confirmed by the user. The weights on the communities can be estimated similarly.

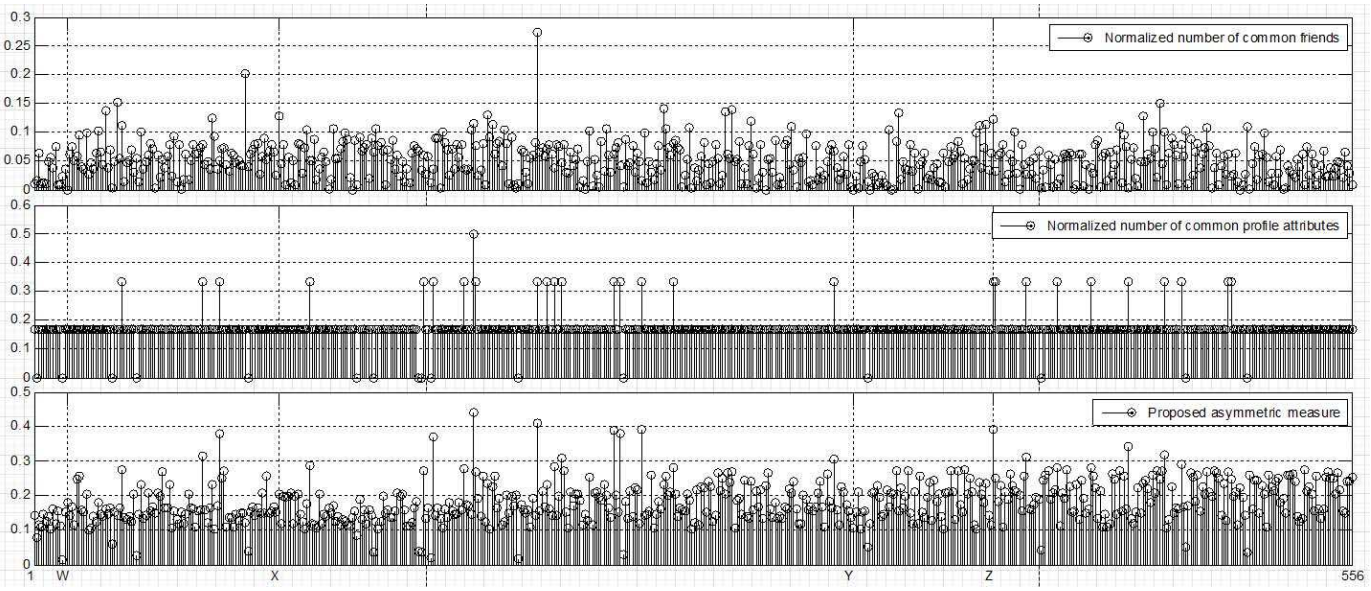


Fig. 3. Calculated Proximity between A and A's friends using Three different Metrics

city T_j communities, the ECE department of university U_i community, and the organization O_i community. The values of α and β are set from 0 to 10. In this experiment, A assigns $\alpha_A^1 = 10, \alpha_A^2 = 9, \alpha_A^3 = 9, \alpha_A^4 = 7, \alpha_A^5 = 5$, and $\alpha_A^6 = 2$. Besides, for all $V \in \overline{\mathcal{N}}_A$ and $C_V^j \in C_V$, we have $\beta_V(C_V^j) = 10$ if $|C_V^j| \geq 50$, and $\beta_V(C_V^j) = 5$ otherwise.

Fig. 3 shows the social proximity values computed based on the normalized number of common friends, the normalized number of common profile attributes, and the asymmetric proximity metric proposed in this paper for each of the 556 nodes (friends of A) in the network. In particular, the normalized number of common friends is calculated as the number of common friends between A and one of A's friends divided by the total number of possible common friends, i.e., 556, in this case. Similarly, the number of common profile attributes (i.e., communities here) is normalized regarding the total number of profile attributes (communities) of A, i.e., $|C_A|$. We contend that compared to the other two metrics, the proposed asymmetric proximity measure can better describe the friendship valuations. In the following, we choose four nodes (W, X, Y, Z) to compare these three metrics in detail.

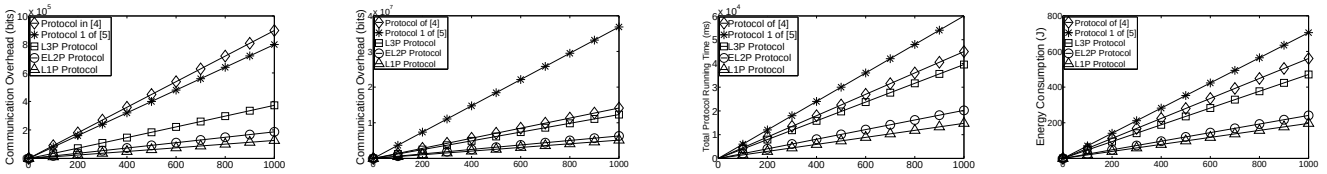
Specifically, the normalized number of common friends cannot fully differentiate the importance of friends. For example, Z and X share approximately the same number of friends with A, and their normalized numbers of common friends with A are 0.11 and 0.13, respectively. In contrast, the proposed asymmetric proximity of Z is nearly twice as much as that of X ($\Psi_{A \leftarrow Z} = 0.39, \Psi_{A \leftarrow X} = 0.20$), since Z shares two communities with A and belongs to two different friend circles FC_A^1, FC_A^2 while X only shares one community with A and belongs to only one friend circle FC_A^1 . The higher social proximity value of Z is justified from the network theory perspective. Particularly, the ratio of *betweenness centrality*[4] of Z to that of X is 5.5 : 1, which emphasizes the relative importance of node Z over X.

Similarly, the normalized number of common attributes fails to well differentiate the importance of friends as well. In our experiment, as shown in Fig. 3, most nodes have the same normalized number of common attributes, and hence cannot be differentiated based on this metric. More importantly, it also fails to fully establish friendships whenever possible. For example, many of A's friends do not have any common attributes with A and hence their normalized numbers of common attributes are 0. On the other hand, the proposed asymmetric proximity measure gives non-zero values as those friends share attributes with some other friends of A. The experiment confirms our argument in the beginning that whether two people can become friends not only depends on whether they have anything (attributes) in common, but also depends on whether their friends have anything in common.

To give another example, friends W and Y have the same normalized number of common friends (0.2) and the same number of common attributes (0). In contrast, the proposed asymmetric proximity measure is able to differentiate these two friends, i.e., 0.18 and 0.11, respectively, as they are associated with different communities and belong to different friend circles with different sizes and weights.

Moreover, we conduct similar experiments on ego-networks of Z, and find that $\Psi_{Z \leftarrow A} = 0.46$ which is larger than $\Psi_{A \leftarrow Z}$, i.e., 0.39, as shown above. Apparently, the results show that Z values the friendship with A more than A does. This is because the size of the overall community set of A is about 15 percent larger than that

4. Betweenness centrality is a measure of a node's centrality in a network [37]. The betweenness centrality of a node v in a network is equal to the number of shortest paths from all nodes to all others that pass through node v.



(a) Total Computation Cost (b) Total Communication Overhead (c) Total Protocol Running Time (d) Total Energy Consumption

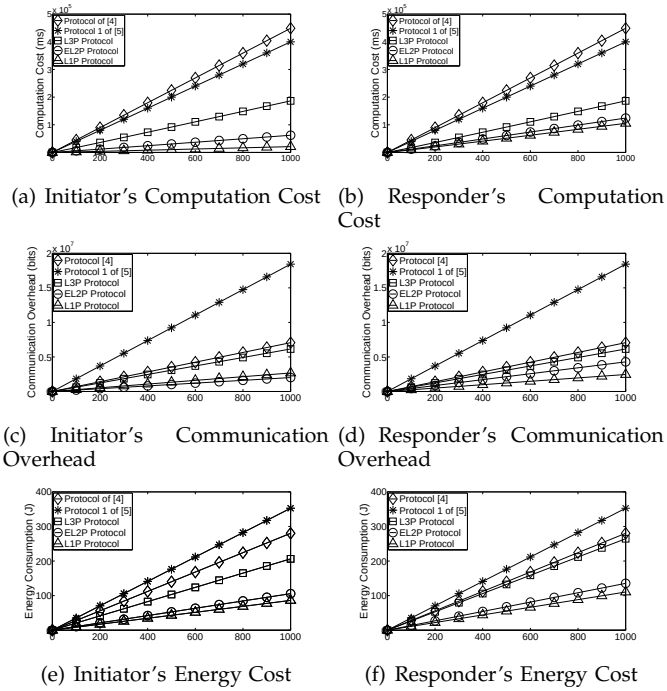
Fig. 4. Comparison of total Computation, Communication, Energy costs, and Protocol Running Time ($|\overline{C}_I| = |\overline{C}_R| = d$ = No. of Friends, $|\overline{C}_I \cap \overline{C}_R| = 10\% \cdot |\overline{C}_I|(|\overline{C}_R|)$ and $\gamma = m = 10$). The X-axis represents the size of overall community set ($|\overline{C}_I| = |\overline{C}_R|$), the size of public profile attribute set (d), and the number of friends in our proposed protocols, [5], and [4] respectively.

of the overall community set of Z ($|\overline{C}_A| = 192, |\overline{C}_Z| = 165$). Besides, A is in two of the total four different friend circles of Z, whereas Z is in two of the six friend circles of A. This demonstrates the asymmetric characteristics of friendships captured by our proximity measure.

5.2 Performance Comparison of Private Matching Protocols

In this section, we evaluate the proposed protocols' performances in terms of computation cost, communication overhead, total running time, and energy cost, and compare them with the performances of the protocols developed in [5] and [4]. In particular, Zhang et al. [5] present fine-grained private matching protocols using an additively separable function like l_1 norm. [5] defines d as the size of the public profile attribute set, which is the set of all possible profile attributes in an OSN. To conduct fair comparisons, we set the size of a user's overall community set equal to d in our proposed protocols. Another parameter γ in [5] denotes the range of the integer used to define a user's level of interest in a particular attribute in the public attribute set. For a reasonably fine-grade private matching, we consider $\gamma = 10$. Besides, [5] presents four protocols with comparable computation and communication complexity. We compare our protocols with their most efficient one: Protocol 1. Besides, Lin et. al [4] propose a privacy preserving friend searching protocol where a user seeks to be introduced to another user's friends with certain attributes. We set attribute size $m = \gamma = 10$ and the number of friends equal to d for fair comparisons. The parameters for the elliptic curve cryptography in [4] are the same as those used in their paper, i.e. we use type D curve of the form $y^2 = x^3 + ax + b$ and the base field is represented by 160 bits.

We have implemented our proposed protocols using a Java implementation of Paillier's cryptosystem [38]. We carry out simulations on a notebook with an Intel Core 2 Duo CPU and 2GB RAM. In the simulations, the same as that in [5], we focus on two wireless nodes communicating with each other, which both use IEEE 802.11 DCF as the MAC protocol with a data rate of 2Mbps. Besides, the energy consumption analysis neglects the energy consumed in computation and only considers the energy cost due to communications. In particular, we



(a) Initiator's Computation Cost (b) Responder's Computation Cost

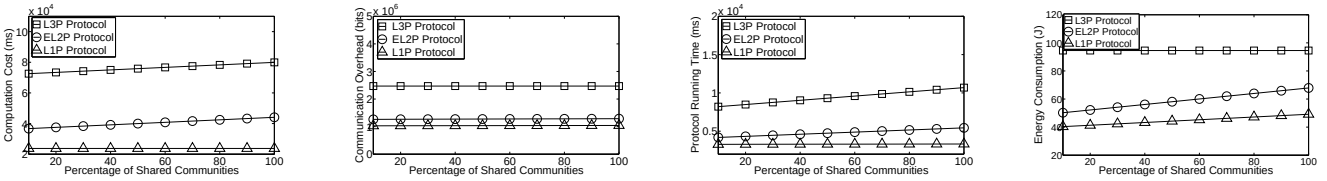
(c) Initiator's Communication Overhead (d) Responder's Communication Overhead

(e) Initiator's Energy Cost (f) Responder's Energy Cost

Fig. 5. Computation, Communication, and Energy cost for the Initiator and for the Responder ($|\overline{C}_I| = |\overline{C}_R| = d$ = No. of friends, $|\overline{C}_I \cap \overline{C}_R| = 10\% \cdot |\overline{C}_I|(|\overline{C}_R|)$ and $\gamma = m = 10$). The X-axis represents the size of overall community set ($|\overline{C}_I| = |\overline{C}_R|$), the size of public profile attribute set (d), and the number of friends in our proposed protocols, [5], and [4] respectively.

follow the energy model in QualNet [39] and assume the *Transmission*: *Reception*: *Idle* energy consumption ratios are 1.57 : 1.14: 1 [40].

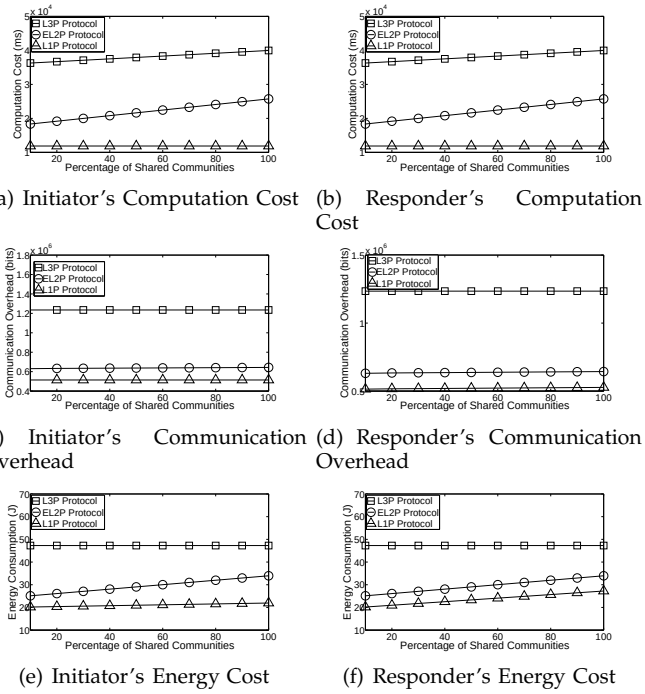
We conduct two sets of simulations in this study. In the first simulation, we vary the size of the overall community set/the public profile attribute set/the number of friends while keeping the percentage of shared community constant at 10%, and γ, m at 10. For simplicity, we consider the Initiator and the Responder have the same overall community set size. Fig. 4(a) compares the total of online and offline computation cost. We can see that our most expensive protocol L3P has much lower computation cost than the most efficient protocol, Protocol 1, of [5] and the protocol of [4]. The



(a) Total Computation Cost (b) Total Communication Overhead (c) Total Protocol Running Time (d) Total Energy Consumption

Fig. 6. Performance comparison with varying size of the percentage of the shared communities between the Initiator and the Responder $|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

reason is that each party in [5] needs to compute $O(d\gamma)$ exponentiations which is very expensive. Similarly, a larger number of ciphertexts due to Zero Knowledge Proof of Knowledge (ZKPoK) and blind key extraction in [4] result in a higher computation complexity. As shown in Fig. 4(b), the communication overhead in [5] and [4] increases faster than our protocols when d/the number of friends increases. The communication cost of [4] is lower than that of [5] because of the smaller size of ciphertexts. Fig. 4(c) compares the total protocol running time. Note that parallel processing between the communication and computation is implemented whenever possible in the protocols. Besides, the total running time takes into account the packet overheads at different layers. In addition, Fig. 4(d) shows the energy consumption of the protocols. We can easily find that our protocols require less running time and consume less energy than Protocol 1 in [5] and the protocol in [4]. We further extend the first experiment by breaking down the computation, communication, and energy costs to those for the Initiator and those for the Responder as shown in Fig. 5. Note that the protocols in [5] and [4] need to run twice in order for both the Initiator and the Responder to obtain the private matching results, and the cost for the Initiator and that for the Responder are the same. We can see that both the Initiator and the Responder are subject to lower costs in our protocols.



(a) Initiator's Computation Cost (b) Responder's Computation Cost

(c) Initiator's Communication Overhead (d) Responder's Communication Overhead

(e) Initiator's Energy Cost (f) Responder's Energy Cost

Fig. 7. Computation, Communication, and Energy cost for the Initiator and for the Responder with varying size of the percentage of the shared communities between the Initiator and the Responder $|\overline{C}_I \cap \overline{C}_R|$ ($|\overline{C}_I| = |\overline{C}_R| = 200$).

In the second set of simulations, we analyze the performance of our protocols when the percentage of common communities between the Initiator and the Responder varies between 0% and 100%. Fig. 6(a) shows the computation cost. In particular, L1P's computation cost is irrelevant to the percentage of common communities since the computation cost of L1P does not depend on $|\overline{C}_I \cap \overline{C}_R|$. Besides, both EL2P's and L3P's computation cost increase only a little as $|\overline{C}_I \cap \overline{C}_R|$ increases. Regarding the communication overhead, Fig. 6(b) shows that the communication overhead of the three protocols almost remains the same even when $|\overline{C}_I \cap \overline{C}_R|$ increases from 0% to 100% of the size of the overall community set. Fig. 6(c) shows the impact of the percentage of common communities on the total protocol running time. EL2P and L3P experience slight increase in total protocol running time since the computational (and communication too in L2P) overhead increases with the increase of $|\overline{C}_I \cap \overline{C}_R|$. Similarly, there is slight increase in the energy consumption of L1P and of EL2P when the fraction of the common communities over the size of the overall community set increases as shown in Fig. 6(d). The energy consumption of L3P remains constant since there is no increase in communication, and hence no additional energy consumption, when the percentage of common communities increase. We further divide the computation, communication, and energy cost in this set of experiments into the corresponding cost incurred by the Initiator and the Responder in Fig 7.

6 Conclusion

The ever increasing use of OSNs has introduced a new paradigm in interacting with existing friends and making new friends in online world as well as in real life. Current schemes lead to privacy breaches. How to enable

people to explore new friends in OSNs while preserving their privacy is an important and challenging problem. In this work, we have exploited the community structure of an OSN to define a realistic asymmetric social proximity measure, and presented three efficient protocols for privately computing the social proximity between two users in OSN. We have validated the proposed measure using real social network data and the simulation study shows the efficacy and the efficiency of the schemes compared to the state-of-the-art schemes.

REFERENCES

[1] (2013, October). [Online]. Available: http://www.alexa.com/topsites

[2] CNN, "Report: Eastern european gang hacked apple, facebook, twitter," http://www.cnn.com/2013/02/20/tech/web/hacked-apple-facebook-twitter/index.html, February, 2013.

[3] IGN, "Microsoft hacked by same method as apple and facebook," http://www.ign.com/articles/2013/02/23/microsoft-hacked-by-same-method-as-apple-and-facebook, February, 2013.

[4] H. Lin, S. S. M. Chow, D. Xing, Y. Fang, and Z. Cao, "Privacy-Preserving Friend Search over Online Social Networks," Cryptology ePrint Archive, Report 2011/445, 2011. [Online]. Available: http://eprint.iacr.org/

[5] R. Zhang, Y. Zhang, J. S. Sun, and G. Yan, "Fine-grained Private Matching for Proximity-based Mobile Social Networking," in *IEEE International Conference on Computer Communications (INFOCOM'12)*, Orlando, Florida, USA, March 2012.

[6] M. Li, N. Cao, S. Yu, and W. Lou, "FindU: Privacy-preserving Personal Profile Matching in Mobile Social Networks," in *IEEE International Conference on Computer Communications (INFOCOM'11)*, Shanghai, China, April 2011.

[7] R. Lu, X. Lin, X. Liang, and X. Shen, "A Secure Handshake Scheme with Symptoms-Matching for mHealthcare Social Network," *Mobile Networks and Applications*, vol. 16, pp. 683–694, 2011.

[8] X. Liang, M. Barua, R. Lu, X. Lin, and X. Shen, "HealthShare: Achieving Secure and Privacy-preserving Health Information Sharing through Health Social Networks," *Computer Communications*, vol. 35, no. 15, pp. 1910–1920, 2012.

[9] W. Dong, V. Dave, L. Qiu, and Y. Zhang, "Secure Friend Discovery in Mobile Social Networks," in *IEEE International Conference on Computer Communications (INFOCOM'11)*, Shanghai, China, April 2011.

[10] H. Zhu, S. Du, M. Li, and Z. Gao, "Fairness-aware and privacy-preserving friend matching protocol in mobile social networks," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 1, pp. 192–200, June 2005.

[11] E. D. Cristofaro, M. Manulis, and B. Poettering, "Private Discovery of Common Social Contacts," in *Proceedings of the 9th international conference on Applied cryptography and network security: ACNS'11*, Nerja, Spain, June 2011.

[12] M. Nagy, E. D. Cristofaro, A. Dmitrienko, N. Asokan, and A.-R. Sadeghi, "Do i know you?: efficient and privacy-preserving common friend-finder protocols and applications," in *Proceedings of the 29th Annual Computer Security Applications Conference*, New Orleans, LA, USA, December 2013.

[13] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient Private Matching and Set Intersection," in *Proceedings of the 17th international conference on theory and application of cryptographic techniques: EUROCRYPT'04*, Interlaken, Switzerland, May 2004.

[14] L. Kissner and D. Song, "Private and Threshold Set-Intersection," in *25th Annual International Cryptology Conference: CRYPTO'05*, Santa Barbara, California, USA, August 2005.

[15] C. Hazay and K. Nissim, "Efficient Set Operations in the Presence of Malicious Adversaries," *Journal of cryptology*, vol. 25, no. 3, pp. 383–433, 2012.

[16] E. D. Cristofaro and G. Tsudik, "Practical Private Set Intersection Protocols with Linear Computational and Bandwidth Complexity," Cryptology ePrint Archive, Report 2009/491, 2009. [Online]. Available: http://eprint.iacr.org/

[17] M. J. Freedman, Y. Ishai, B. Pinkas, and O. Reingold, "Keyword Search and Oblivious Pseudorandom Functions," in *Proceedings of the Second international conference on Theory of Cryptography*, Cambridge, MA, USA, 2005.

[18] C. Hazay and Y. Lindell, "Efficient Protocols for Set Intersection and Pattern Matching with Security against Malicious and Covert Adversaries," in *Proceedings of the 5th conference on Theory of cryptography*, New York, USA, 2008.

[19] S. Jarecki and X. Liu, "Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection." in *Theory of Cryptography: TCC'09*, San Francisco, CA, USA, March 2009.

[20] J. Camenisch and G. M. Zaverucha, " Private Intersection of Certified Sets," in *Financial Cryptography and Data Security*, Accra Beach, Barbados, February 2009.

[21] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy, "Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data," in *Proceedings of the 12th International Conference on Practice and Theory in Public Key Cryptography: PKC '09*, Irvine, CA, USA, March 2009.

[22] E. D. Cristofaro, J. Kim, and G. Tsudik, "Linear-Complexity Private Set Intersection Protocols Secure in Malicious Model." in *16th International Conference on the Theory and Application of Cryptology and Information Security: ASIACRYPT'10*, Singapore, December 2010.

[23] A. Yao, "How to Generate and Exchange Secrets," in *27th Annual Symposium on Foundations of Computer Science*, October 1986.

[24] O. Goldreich, S. Micali, and A. Wigderson, "How to play ANY mental game," in *Proceedings of the nineteenth annual ACM symposium on Theory of computing*, New York, USA, 1987.

[25] I. Damgård, Y. Ishai, and M. Krøigaard, "Perfectly Secure Multiparty Computation and the Computational Overhead of Cryptography," in *Proceedings of the 29th Annual international conference on Theory and Applications of Cryptographic Techniques: EUROCRYPT'10*, French Riviera, France, May 2010.

[26] I. Damgård, Y. Ishai, M. Krøigaard, J. B. Nielsen, and A. Smith, "Scalable Multiparty Computation with Nearly Optimal Work and Resilience," in *Proceedings of the 28th Annual conference on Cryptology: Advances in Cryptology*, Santa Barbara, CA, USA, 2008.

[27] Z. Beerliová-Trubíniová and M. Hirt, "Perfectly-secure MPC with linear communication complexity," in *Proceedings of the 5th conference on Theory of cryptography*, New York, USA, 2008.

[28] M. Franklin and M. Yung, "Communication Complexity of Secure Computation (extended abstract)," in *Proceedings of the twenty-fourth annual ACM symposium on Theory of computing: STOC'92*, Victoria, British Columbia, Canada, May 1992.

[29] R. Gennaro, M. O. Rabin, and T. Rabin, "Simplified VSS and Fast-track Multiparty Computations with Applications to Threshold Cryptography," in *Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing*, Puerto Vallarta, Mexico, June 1998.

[30] M. Hirt, U. M. Maurer, and B. Przydatek, "Efficient Secure Multiparty Computation," in *Proceedings of the 6th International Conference on the Theory and Application of Cryptology and Information Security: ASIACRYPT'00*, Kyoto, Japan, December 2000.

[31] L. Katz, "A New Status Index Derived from Sociometric Analysis," *Psychometrika*, vol. 18, 1953.

[32] D. Liben-Nowell and J. Kleinberg, "The Link Prediction Problem for Social Networks," in *Proceedings of the twelfth international conference on Information and knowledge management*, New Orleans, LA, USA, 2003.

[33] H. Tong, C. Faloutsos, and Y. Koren, "Fast Direction-aware Proximity for Graph Mining," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*, New York, NY, USA, 2007.

[34] L. Zhang, X.-Y. Li, and Y. Liu, "Message in a sealed bottle: Privacy preserving friending in social networks," in *IEEE 33rd International Conference on Distributed Computing Systems (ICDCS'13)*, Philadelphia, USA, July 2013.

[35] (2013, July). [Online]. Available: http://www.diasporaproject.org/

[36] P. Paillier, "Public-key Cryptosystems based on Composite Degree Residuosity Classes," in *Proceedings of the 17th international conference on theory and application of cryptographic techniques: EUROCRYPT'99*, Prague, Czech Republic, 1999.

[37] L. C. Freeman, "A Set of Measures of Centrality Based on Betweenness," *Sociometry*, vol. 40, pp. 35–41, 1977.

[38] K. Liu, "Paillier's cryptosystem in java," http://www.csee.umbc.edu/ kunliu1/research/Paillier.html.
[39] http://web.scalable-networks.com/content/qualnet.
[40] Q. . W. M. Library, http://rainbow.sunmoon.ac.kr/qualnet/menuals/Documents/ModelLibraries/QualNet-5.1-Wireless-ModelLibrary.pdf.

Arun Thapa received his B.E. degree in Electronics and Communication Engineering from National Institute of Technology, Durgapur, West Bengal, India, in 2005. He is currently working towards his Ph.D. degree in the Department of Electrical and Computer Engineering, Mississippi State University. Prior to pursuing his Ph.D. at Mississippi State University, he was a Telecom Engineer in Nepal Telecom. His research interests include security and privacy in wireless networks and social networks. He is a student member of the IEEE.

Ming Li received the B.E. degree in Electrical Engineering from Sun Yat-sen University, China, in 2007, and the M.E. degree in Electrical Engineering from Beijing University of Posts and Communications, China, in 2010, respectively. She is currently working towards her Ph.D. degree in the Department of Electrical and Computer Engineering, Mississippi State University. Her research interests include cross-layer optimization, and security and privacy in cognitive radio networks, cloud computing, and smart grids. She is a student member of the IEEE.

Sergio Salinas received the B.S. degree in telecommunications engineering from Jackson State University, Jackson, in 2010. He is currently working towards his Ph.D. degree in the Department of Electrical and Computer Engineering, Mississippi State University. His research interests include cyber-physical systems, cloud computing, and online social networks. He is a student member of the IEEE.

Pan Li received the B.E. degree in Electrical Engineering from Huazhong University of Science and Technology, Wuhan, China, in 2005, and the Ph.D. degree in Electrical and Computer Engineering from University of Florida, Gainesville, in 2009, respectively. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering, Mississippi State University. His research interests include network science and economics, energy system, security and privacy, and big data. He has been serving as an Editor for IEEE Journal on Selected Areas in Communications – Cognitive Radio Series and IEEE Communications Surveys and Tutorials, a Feature Editor for IEEE Wireless Communications, a Guest Editor for IEEE Wireless Communications SI on User Cooperation in Wireless Networks, and a TPC Co-Chair of Wireless Networking Symposium, IEEE ICC 2013. He received the NSF CAREER Award in 2012 and is a member of the IEEE and the ACM.