Weight-Aware Private Matching Scheme for Proximity-based Mobile Social Networks

Ben Niu⁵, Xiaoyan Zhu⁴, Jie Liu‡, Zan Li†, Hui Li‡
National Key Laboratory of Integrated Networks Services,
Xidian University, Xi’an 710071, China
Email: ⁵ xdnuben@gmail.com, ⁶ (xyzhu, zanli, lihui)@mail.xidian.edu.cn, ⁷ liujie779@163.com

Abstract—Making new social interactions with other users in vicinity is a crucial service in Proximity-based Mobile Social Networks (PMSNs), where a user can find a best matching friend directly through the Bluetooth/WiFi interfaces built in her mobile device. In existing work for such services, users have to publish their interests to do the matching. However, it conflicts with users’ growing privacy concerns about revealing their interests to strangers. To tackle this problem, we propose Weighted Average Similarity (WAS) algorithm, which considers both the number of common interests and the corresponding weights on them, to protect users’ privacy without reliance on any Trusted Third Party (TTP). Users set their interests into several priority levels with different weights, then WAS can provide a high level similarity value among these participants without revealing any information about their common interests. The security and computation/communication overhead of our scheme are thoroughly analyzed and evaluated via detailed simulations.

Index Terms—Private Matching; Privacy-Preserving; PMSNs

I. INTRODUCTION

With the explosive growth of mobile devices, such as smartphones and tablets, Mobile Social Networks (MSNs) have become a vital part in our daily life. MSNs enable users not only to enjoy their existing social network applications (e.g., Facebook, Foursquare, etc.) anywhere and anytime, but also to participate in the emerging techniques such as Location-based Services (LBSs). Among these applications, an important service is Proximity-based Mobile Social Networks (PMSNs), which refers to the social interaction among physically proximate mobile users directly through the Bluetooth/WiFi interfaces on their smartphones. PMSNs provide users more chances to make new social interactions with their neighbors, and they are the only way to do social networking when there is lack of Internet access, e.g., due to the very weak signals from cellular base station or some natural disaster scenarios like earthquake. To enjoy these activities, the first step is to choose potential friends from nearby users. A straightforward way is that an initiator broadcasts her interests directly to all the nearby users, and then the responder makes decision according to the common interests. This scheme works well but conflicts with the mobile users’ privacy. Since they would not want to reveal their interests to strangers before deciding to interact with them.

To address this problem, many existing work have been proposed in recent years. They can be divided into two main categories: TTP-based approaches [1], [2], [3], [4], [5], [6] and Private Set Intersection (PSI)-based approaches [7], [8], [9], [10], [11]. In TTP-based approaches, users submit their interests to the TTP, which acts as a matching center to compute the similarity between each pair of users, and replies each user with the best match. Unfortunately, in this kind of servers, TTP is the bottleneck from both the security and system performance points of view. The reason is that the TTP needs to know all the users’ interests to perform the matching process, so it is quite dangerous when the TTP is compromised. PSI-based approaches suffer from attacks which are caused by the unlimited input, since users can freely input their interests. As a result, an attacker can perform attacks by inputting as many interests as he can. To cope with the aforementioned problems, authors in FindU [11] set a limitation to the number of interests. In their paper, they limit this number to 200. However, it is hard to verify the rationality of this number. As an improvement to PSI, De Cristofaro et al. [12] propose the authorized PSI, which certifies each interest before being sent to other entities, however, it also relies on a TTP to do the certification work. We also consider the matching results of existing schemes, many of which measure the similarity between users by counting the number of common interests [11], [13], large number of common interests always indicates higher probabilities to be friends [14]. However, this situation conflicts with the fact in our daily life, since users always have many different interests, and with different priorities on them, which means users may pay much attention to some of the interests, and pay little attention on the others. From this point of view, approaches based on counting the number of common interests do not work well. To cope with this issue, Zhang et al. [15] consider the priorities on each common interest, and define several privacy levels to realize the fine-grained private matching schemes in their work, but they only pay attention to the difference of priorities on each common interest among users, but ignore the priority itself. We illustrate our concerns with a common example in Fig. 1, in which the shaded cells means the interests with more interesting, while the others represent the normal interests. The user Alice wants to find the best match in vicinity. In this example, it is clear that the best match of Alice is Bob instead of Charles, even though there are three common interests...
between Alice and Charles and only one with Bob.

Based on the aforementioned analysis, we can conclude two main drawbacks of existing work: 1) reliance on trusted third party; 2) ignorance on the weights associated with interests. In this paper, we propose a novel and efficient privacy-persevering friend discovery scheme for the increasingly popular Proximity-based Mobile Social Networks. The main contributions of this paper are shown as follows.

- We propose a novel and TTP-free algorithm, Weighted Average Similarity (WAS), to measure the similarity between users. WAS considers both the number of common interests and the related weights on them. The main idea of our scheme is to assign interests into different levels in terms of weights, and make comparison between levels instead of each interest individually to protect user’s privacy.

- We implement our scheme and evaluate the performance. The evaluation results indicate that our scheme saves much execution time as well as the energy consumptions.

The rest of this paper is organized as follows. Section II shows some preliminaries in our work. Following in Section III, we describe the system architecture and the details of our proposed system. Sections IV and V show the security analysis and our evaluation results. Finally, we draw the conclusions in Section VI.

II. PRELIMINARIES

A. Motivation and Our Main Idea

Our work is motivated by the emerging applications on friend discovery, which help users to make new social interactions efficiently by the widely used smartphones. In our daily life, we have many chances to make new friends with others in vicinity, and if these chances can be hold, they may bring us lots of fan (share and obtain more interesting information with others) and benefits (gain and seize more business opportunities). The Proximity-based Mobile Social Networks (PMSNs) provide us a good platform to do social networking with nearby people even we do not have Internet connections. Through smartphones, users can communicate and learn the common interests with others easily to decide if they can become friends. Intuitively, they can share their interests to each other to know the similarity between them, which is efficient but at the cost of users’ privacy. To address the privacy issue, many approaches have been proposed recently. However, most of them [11] establish the relationship of friend when two users have more common interests (e.g., when the number of common interests surpasses the predefined threshold). Authors in [15] overcome this problem by considering the weight on each interest. They compute the sum of the differences on each interest to measure the similarity. Based on the complicated cryptographic tools such as secure multiple computation, both of the two aforementioned solutions can protect users’ privacy. However, they ignore some common senses: 1) the number of common interests can not be the only judgement condition to make relationships with others; 2) for each common interest, users may assign different weights on it. For example, suppose Alice and Bob have a common interest Thai Food, the weights (between 0 to 9, bigger value means more interesting on it) they assign to them are 9 and 8, respectively. On the other hand, Alice and Charles have another common interest Tennis, the corresponding weights are 2 and 1, respectively. Obviously, Alice wants to make friends with Bob instead of Charles even the difference on Thai Food equals to that on Tennis (9 - 8 = 2 - 1).

To address these problems, our general idea is to use several priority levels to assign user’s interests. We set different weights to different priority levels. In the matching phase, instead of computing the differences or similarities on each common interest individually, we introduce the weighted average similarity, which compares the similarity between different priority levels. It considers both the number of the common interests and the weights assigned to them, and shows a high level similarity value to participants. As a result, each participant knows the weighted average similarity but nothing about each common interest.

B. Problem Statement

Our scheme can be a social networking application of PMSNs, which involves several users and with no reliance on TTP. The whole process can be divided into three phases: the linking phase, the matching phase, and the information exchanging phase. Our work focuses on the first and second phases. We use two users Alice and Bob for example. Alice is with profile \( I_A = \langle I_{A_1}, I_{A_2}, \cdots \rangle \), where \( I_{A_i} \) may be interests such as movie, music, football and so on. While Bob has a profile as \( I_B = \langle I_{B_1}, I_{B_2}, \cdots \rangle \). Users always have different priorities on their interests, e.g., Alice may love movie very much but less interesting on football. Suppose there is a predefined number to divide all the interests into several priority levels (between 1 to \( l \)), in which the bigger priority level indicates the interest with more concerns. Then all the interests of Alice or Bob can be categorized into the \( l \) levels. We use a Weighted Matching Method to measure the similarity between users. \( x_i \) represents the number of common interests within any two levels of two users, respectively. \( w_i \) means the corresponding weight on each priority level. For Alice in the matching phase, she computes the weighted priority value as

\[
\pi = \frac{\sum_{i=1}^{l} x_i \cdot w_i}{\sum_{i=1}^{l} w_i},
\]
which indicates the weighted average similarity between common interests of two users. Based on the weighted priority value, a user can learn the degree of similarity with others but knows nothing about the common interests individually.

C. Adversary Models

Normally, each user in our scheme is honest-but-curious, which means they try to learn more information than allowed about other users. However, there are something different between the initiator and the responder. On the initiator side, she can also be an attacker directly, since she may illegally input her interests and the related weights to learn more information of nearby users. While on the responder, she can be either a honest-but-curious or legitimate user. We do not worry about the responder would be an attacker since the identity of a responder can be easily authorized by other entity. Moreover, our scheme can be easily reversed to achieve a dual matching.

D. Design Goals

Our main goal is to achieve a secure private matching between an initiator and several responders. After running our protocol, both the initiator and responder can know a full view of the similarity degree, which indicates not only the number of common interests but also the weights related to them. Users cannot infer any related information of each individual interest from the obtained similarity value.

E. Cryptographic Tool

We realize our idea by utilizing the Commutative Encryption Function [16], [17], which satisfies the condition: $E_{k_1}(E_{k_2}(x)) = E_{k_2}(E_{k_1}(x))$. In our protocol, it indicates that two users, who have the secret keys $k_1$ and $k_2$, respectively, can know that $x_1 = x_2$ iff. $E_{k_1}(E_{k_2}(x_1)) = E_{k_2}(E_{k_1}(x_2))$. More specifically, we choose the power function $f_k(x) = x^k \mod p$ as the commutative encryption in our protocol. $p$ and $p-1$ are two safe primes. For all integers $k_1, k_2$ and $x \in \mathbb{Z}_p^*$, there exists an integer $n$, s.t.

$$f_{k_1}(f_{k_2}(x)) \equiv f_{k_1}(x^{k_2} \mod p) \equiv f_{k_1}(x^{k_2 - np}) \equiv (x^{k_2 - np})^{k_1} \mod p$$

$$\equiv x^{k_2 k_1 \mod p} \equiv f_{k_2}(f_{k_1}(x)). \quad (2)$$

To simplify the calculation, we need a hash function to work as the original interests. A hash function is used to map a larger data set to a smaller one with a fixed length.

III. SYSTEM ARCHITECTURE AND OUR PROPOSED PROTOCOL

A. System Architecture

Based on their different roles, we can roughly divide entities into two categories: certificate authority (CA) and mobile users. Mobile users can be further classified into initiator and responder. Specifically, CA is responsible for generating and managing certificates of mobile users. An initiator is a user who wants to initiate a friend discovery request with nearby users. A responder is a user who tries to reply the received requests. These mobile users have abilities to communicate with CA through cellular networks, such as 3G/4G. Meanwhile, they use Bluetooth to communicate with adjacent users since it consumes less power. We did not employ any other trusted third party into our scheme to avoid bottlenecks on both security concerns and system performance.

B. Our Proposed Protocol

In this subsection, we present our proposed Weighted Average Similarity algorithm in Algorithm 1 in details. The initiator and responder hold the profiles $I_A$ and $I_B$, respectively. They need to assign these interests into $l$ levels with different weights and each level may involve a single or several interests. The secret keys of the initiator and responder are $k_A$ and $k_B$, respectively. $\delta_i$ represents the weighted average similarity between the $i$th level of the initiator and all the $l$ levels of the responder. $\tau$ is the weighted average similarity between the initiator and responder. In our algorithm, we employ an array comparison function IsEqual() which can compute the number of common elements between two arrays. All the results are $\mod p$ in our algorithm.

\textbf{Algorithm 1} Algorithm on Responder (B) side

\textbf{Input:} $(h(I_{B_i})^{k_B})^{k_A}, (h(I_A)^{k_A})^{k_B}, i = 1, 2, \cdots, l$;

\textbf{Output:} $\tau$

1: for $(i = 1; i \leq l; i++)$ do
2: for $(m = 1; m \leq l; m++)$ do
3: $N_{im} = \text{IsEqual}((h(I_A)^{k_A})^{k_B}, (h(I_{B_i})^{k_B})^{k_A})$
4: end for
5: end for
6: for $(i = 1; i \leq l; i++)$ do
7: for $(m = 1; m \leq l; m++)$ do
8: $\text{Sum}_i = N_{im} \ast (l - |i - m|) + \text{Sum}_i$;
9: end for
10: $\delta_i = \frac{\text{Sum}_i}{l}$;
11: end for
12: $\tau = \sum_{i=1}^{l} (\delta_i \ast i)$;

(i) The initiator computes $h(I_{A_i})$ and $(h(I_A)^{k_A})^{k_B}$ by using her secret key $k_A$ and broadcasts these information to others for replies;
(ii) The responder replies $(h(I_{B_i})^{k_B})^{k_A}$ to the initiator by computing $h(I_{B_i})$ and $(h(I_{B_i})^{k_B})^{k_A}$ in turn;
(iii) Upon these receiving messages, the initiator then computes $(h(I_{B_i})^{k_A})^{k_B}$ and sends them back to the responder;
(iv) The responder computes $(h(I_A)^{k_A})^{k_B}$ for comparison, he compares $(h(I_A)^{k_A})^{k_B}$ with the received $(h(I_{B_i})^{k_A})^{k_B}$. If they are equal, he records the number of the common interests between two levels and we denote it as $N_{im}$. Specifically, $N_{im}$ is the number of common
interests between the ith level of the initiator and the mth level of the responder. He then computes the weighted average similarity $\delta_i$ between the ith level of a user with all the l levels of the other user. It can be computed by:

$$\delta_i = \frac{(N_{i1} \times (l - |i - 1|)) + \cdots + (N_{il} \times (l - |i - l|))}{l}.$$  

(3)

Based on these computed $\delta_i$s, the algorithm can further get the weighted average similarity $\tau$ between two users:

$$\tau = \sum_{i=1}^{l} (\delta_i \times w_i) = \sum_{i=1}^{l} (\delta_i \times i),$$  

(4)

and reply the initiator with $\tau$.

(v) At last, according to the received value of weighted average similarity $\tau$, the initiator decides whether to make friends with the responder.

IV. SECURITY ANALYSIS

We provide some security proofs based on our adversary models, including resistance to attacks from outside adversary, and privacy protections for the initiator and the responder.

A. Resistance to attacks from outside adversary

Any entity can be an outside adversary if he can monitor and eavesdrop on the wireless channel between entities. However, in our scheme, cryptography techniques such as the Public Key Infrastructure (PKI) are used, where all the entities need to register certificates with CA, and they can check other entities by verifying the certificates as well as exchange ciphertext encrypted by their secret keys with others. Through this way, attacks such as eavesdropping attacks on the wireless channel can be avoided.

B. Privacy of the initiator

The responder always learns more information than the initiator does. In our scheme, the received messages on the responder side include $h(I_{Ai})^{k_B}$ and the computed $\tau$.

**Theorem 1:** The privacy of the initiator can be protected if the commutative encryption function is secure.

**Proof:** The initiator reveals all the interests $h(I_{Ai})^{k_A}$ to the potential responders in vicinity, however, all these interests are encrypted under her secret key $k_A$. In our scheme, the commutative encryption function and the hash function ensure that the responder can only obtain the common interests, since $(h(I_{Ai})^{k_A})^{k_B} = (h(I_{Bi})^{k_B})^{k_A}$ if and only if 1): they have common interests $I_{Ai} = I_{Bi}$; 2) each of the initiator and the responder holds one of the encryption keys. Through this way, the initiator’s privacy is protected from both the outside adversary and the responder.

C. Privacy of the responder

The received messages on the initiator side include $h(I_{Bi})^{k_B}$ and $\tau$ from responder side, however, the former message is encrypted by the responder’s secret key $k_B$. It is computational hard for the initiator to obtain the interests. The initiator can also get the weighted average similarity value $\tau$.

Generally speaking, bigger $\tau$ indicates a higher similarity, and it is impossible to deduce any useful information, such as the number of common interests.

**Theorem 2:** In our scheme, the initiator can confirm a common interest of the responder iff. she has only one interest which is also the only one common interest. Otherwise, the initiator would know nothing if she has at least two common interests with the responder.

**Proof:** When the initiator and responder have only one common interest, the initiator can confirm that the responder also has this interest if the received $\tau \neq 0$. However, since we compute $\tau$ based on a dual weighted average function, so the initiator cannot get other related information about the only one common interest, such as the weight that the responder assigns on this interest. Suppose the initiator has two or more interests, then let us recall the Equations 3 and 4. It is hard to determine the total number of common interests with the responder as well as the number of the common interests $N_{im}$ with each level of the responder, as the result, the initiator learns nothing.

V. PERFORMANCE ANALYSIS

In this section, we verify the feasibility of our scheme, analyze the complexity of the protocol, carry out lots of simulation study of the protocols efficiency, and also compare it with several related schemes in terms of security and efficiency.

A. Experiment of Correctness

We use a simple example to verify the correctness of our scheme. In Table I, we assume four users Alice, Bob, Charles and David are within the communication range with each other. Suppose Alice is the initiator and she broadcasts a query to others. Bob, Charles and David are the responders, who receive and respond the query. For simplicity, we use extreme, normal and little to describe the different levels the users assign their interests. We limit the maximum number of interests to 15. $I_i$ is the ith level of an interest that a user is interested in. We then compare the matching results of some existing work with our scheme.



| TABLE I | EXPERIMENT SETTING |
|---------|-------------------|
| Alice   | $I_1, I_3, I_5, I_8$ | $I_6, I_{10}, I_{13}$ | $I_2, I_4, I_7, I_9$ |
| Bob     | $I_1, I_3, I_5, I_8$ | $I_4, I_{12}, I_{14}, I_{15}$ | $I_2, I_6, I_{10}$ |
| Charles | $I_1, I_3, I_8$ | $I_6, I_{10}, I_{13}$ | $I_2, I_4, I_7, I_9$ |
| David   | $I_2, I_6, I_7, I_9$ | $I_1, I_3, I_4, I_8$ | $I_5, I_{10}, I_{13}$ |

In the **Level-I privacy** proposed in [15], they use the Manhattan Distance ($l_1$-distance) matching metric to measure the similarity between two vectors according to the equation $l = \sum_{i=1}^{m} |I_{Ai} - I_{Bi}|$, $m$ means the number of the common interests, and $l_i$ represents the weight assigned to each common interest. It is easy to understand that a smaller $l$ leads to a better match. As the matching result in [15], the best match
is Charles since the \( \ell_1 \)-distance between Alice and Charles is 12, which is the smallest value compared with 14 to Bob and 16 to David. While in FindU [11], they just consider the number of the common interests among users. The output is David since they have 11 common interests.

In our scheme, we set different weights to different priority levels instead of particular interests. We compute the weighted average similarity \( \tau \) of all the interests. The bigger \( \tau \) indicates the better match result. The weighted average similarity between Alice and Bob is 18.6667. While the result with Charles is 16.3350, and 15.0000 with David. According to our scheme, we know the best match of Alice is Bob.

### B. Complexity Analysis

In this subsection, we evaluate the complexity of our scheme and compare with some existing work. The complexity is measured from offline, online computation cost and execution time as well as the communication overhead. The computation cost is measured by counting hash function, modular exponentiations and multiplications, since these operations are always resource-consuming in mobile devices. \( h \) represents the hash function SHA-256, and \( \text{Sig} \) refers to the RSA signature with a modulus length of 1024 bits and \( \text{exp}_1 \) means 1024-bit exponentiation operations. We assume that each user in our scheme has \( m \) interests. The communication cost is evaluated by computing the transmitted and received bits. We then compare our work with [17], [12] shown in Table II, since the scheme in [17] uses the commutative encryption in privacy matching phrase as our work. However, they use a third party to do the offline work, such as generating the parameters. Scheme in [12] considers the malicious behavior in private matching as our work. From this table, we can see that our scheme saves the computation and communication cost significantly, especially the online computation cost, which affects the system performance directly.

#### C. Simulation Setup

In order to verify the feasibility of our protocol, we implement our scheme in C++ language running on a 2.94 GHZ Windows 7 system and the cryptography library is Crypto++. Every data point in our experiments is an average of 1000 runs. We ignore the impacts of some simple operations, such as multiplication operations, since they need less computation cost compared with exponentiation operations.

#### D. Simulation Results

Our experiments show that we need 178.4332 ms to generate a safe prime \( p \) with 1024 bits which can be computed offline. The \( \text{exp}_1 \) needs 0.3298 ms. The RSA signatures generating phase needs 9.1227 ms, and the SHA-256 requires 0.0019 ms.

Fig. 2(a) shows the relationship between offline computation cost and the number of interests \( m \). Our scheme consumes less computation cost than [12], which uses too much \( \text{exp}_1 \)s to achieve their goal. While in [17], there is no offline computation cost, since they are done on a third party, called Personal Interest Signer (PIS).

Fig. 2(b) compares the online computation cost of all the protocols in the log10 scale for a varying \( m \). It’s clearly that our scheme is more efficient than others, since we only need \( m \cdot \text{exp}_1 \) on both the initiator side and responder side. The online computation cost is sensitive to the whole execution time, which is closely related to the user’s experience. Scheme in [17] works worse by the reason of more exponentiation operations and extra RSA signatures.

Fig. 3(a) shows the total protocol execution time with varying \( m \), from 20 to 200. This experiment is quite useful to measure the usability of a scheme. Comparing with other schemes, our scheme can be finished within about 200 ms in all the tested scenarios. For example, when the number of interests \( m = 200 \), algorithms in [12] and [17] need 15.3991 and 4.4131 seconds to complete the matching phase for each user, while our scheme requires only 183.3041 ms.

Fig. 3(b) shows the comparison results of communication cost among these protocols. It includes both the transmitting bits and receiving bits. Obviously, each algorithm increases al-

### Table II

| Party      | Our protocol | [12] | [17] |
|------------|--------------|------|------|
| Offline Comp. | Initiator | \( m \cdot \text{exp}_1 + m \cdot h \) | \( (2m + 2m^2) \cdot \text{exp}_1 + (2m) \cdot h \) | — |
|            | Responder   | \( m \cdot \text{exp}_1 + m \cdot h \) | \( (m + m^2) \cdot \text{exp}_1 + (2m) \cdot h \) | — |
| Online Comp. | Initiator | \( m \cdot \text{exp}_1 \) | \( (m + m^2) \cdot \text{exp}_1 + (m) \cdot h \) | \( (2m) \cdot \text{exp}_1 + (m + 2) \cdot \text{Sig} + h \) |
|            | Responder   | \( m \cdot \text{exp}_1 \) | \( 2m \cdot \text{exp}_1 \) | — |
| Comm.bits  | Initiator   | \( 2m \cdot 1024 \) | \( 4m \cdot 1024 \) | \( (6m + 3) \cdot 1024 \) |
|            | Responder   | \( (m + 2) \cdot 1024 \) | \( 3m \cdot 1024 \) | \( (6m + 3) \cdot 1024 \) |
|            | Two party   | \( (3m + 2) \cdot 1024 \) | \( 7m \cdot 1024 \) | \( (12m + 8) \cdot 1024 \) |
most linearly with increasing $m$. For example, when $m = 100$, our scheme only needs to consume $0.2553 \text{Mb}$ on bandwidth. It is easy for Bluetooth 3.0 and 4.0, which have a promised speed of $25 \text{Mb/s}$. In the same case, schemes in [12] and [17] consume $0.4371 \text{Mb}$, and $0.8742 \text{Mb}$, respectively.

**E. Energy Consumption**

We also compute the energy consumption of our scheme and others. The energy consumption model is based on the formula: $E = n_t \cdot E_t + n_r \cdot E_r$ in [18], where $n_t$ and $n_r$ are the transmitted and received data in bytes, and $E_t \approx 4.85 \mu J$ is transmitting energy per byte, $E_r \approx 6.7\mu J$ is the receiving energy per byte. For simplicity, we omit the initial connection establishment energy since it is common in all schemes.

Fig. 4 shows the comparison of energy consumption on each side of the protocols. Our scheme improves the energy consumption significantly. When $m = 200$, the initiator in our protocol needs $1.8380 J$, and the responder requires $1.9343 J$. Scheme in [12] needs $4.2012 J$ on the initiator side, and $4.1094 J$ on the responder. The protocol in [17] consumes $28.3527 J$ on the initiator, and $28.3600 J$ on the responder side.

**VI. CONCLUSIONS**

In this paper, we proposed a privacy-preserving friend discovery for Proximity-based Mobile Social Networks with no reliance on any trusted third party. Comparing with existing work, our proposed Weighted Average Similarity algorithm computes the similarity between two users by considering both the number of the common interests and the corresponding weights on them. We let users to assign their interests into levels with different weights, to protect user’s privacy. WAS computes a high level similarity between different levels of users instead of each common interest individually. Security analysis shows that our proposed WAS can protect users’ privacy on both sides of the *initiator* and *responder*. Finally, evaluation results indicate that WAS can achieve a better matching correctly and efficiently.

**ACKNOWLEDGEMENT**

This work was partially supported by National Natural Science Foundation of China under Grant 61003300, Fundamental Research Funds for the Central Universities under Grant K5051201041, and China 111 Project under Grant B08038. The work of Dr. Hui Li was partially supported by National Natural Science Foundation of China under Grant 61272457. The work of Dr. Zan Li was partially supported by Major National Science & Technology Projects under Grant 2010ZX03006-002-04.

**REFERENCES**

[1] N. Eagle and A. Pentland, “Social serendipity: mobilizing social software,” IEEE Pervasive Computing, vol. 4, no. 2, pp. 28 – 34, jan.-march 2005.
[2] L. P. Cox, A. Dalton, and V. Marupadi, “Smokescreen: flexible privacy controls for presence-sharing,” in ACM MobilSys 2007.
[3] J. Manweiler, R. Scudellari, and L. P. Cox, “Smile: encounter-based trust for mobile social services,” in ACM CCS 2009.
[4] A.-K. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, “Mobiclique: middleware for mobile social networking,” in ACM WOSN 2009.
[5] S. Rane, W. Sun, and A. Vetro, “Privacy-preserving approximation of 11 distance for multimedia applications,” in IEEE ICME 2010.
[6] W. Dong, V. Dave, L. Qiu, and Y. Zhang, “Secure friend discovery in mobile social networks,” in IEEE INFOCOM 2011.
[7] M. J. Freedman, K. Nissim, and B. Pinkas, “Efficient private matching and set intersection,” in EUROCRYPT 2004.
[8] L. Kissner and D. Song, “Privacy-preserving set operations,” in CRYPTO 2005.
[9] Y. Sang and H. Shen, “Efficient and secure protocols for privacy-preserving set operations,” ACM Transactions on Information and System Security (TISSEC), vol. 13, no. 1, pp. 9:1–9:35, Nov. 2009.
[10] G. Ateniese, E. De Cristofaro, and G. Tsudik, “(if) size matters: size-hiding private set intersection,” in ACM PKC 2011.
[11] M. Li, N. Cao, S. Yu, and W. Lou, “Findu: Privacy-preserving personal profile matching in mobile social networks,” in IEEE INFOCOM 2011.
[12] E. De Cristofaro, J. Kim, and G. Tsudik, “Linear-complexity private set intersection protocols secure in malicious model,” in ASIACRYPT 2010.
[13] Z. Yang, B. Zhang, J. Dai, A. Champion, D. Xuan, and D. Li, “E-smalltalker: A distributed mobile system for social networking in physical proximity,” in IEEE ICDCS 2010.
[14] X. Liang, X. Li, K. Zhang, R. Lu, X. Lin, and X. Shen, “Fully anonymous profile matching in mobile social networks,” IEEE Journal on Selected Areas in Communications (JSAC), to appear, 2013.
[15] R. Zhang, Y. Zhang, J. Sun, and G. Yan, “Fine-grained private matching for proximity-based mobile social networking,” in IEEE INFOCOM 2012.
[16] R. Agrawal, A. Evfimievski, and R. Srikan, “Information sharing across private databases,” in ACM SIGMOD 2003.
[17] Q. Xie and U. Hengartner, “Privacy-preserving matchmaking for mobile social networking secure against malicious users,” in IEEE P2P 2011.
[18] A. Rahmati and L. Zhong, “Context-for-wireless: context-sensitive energy-efficient wireless data transfer,” in ACM Mobisys 2007.