Big Data Research ••• (••••) •••–•••

Contents lists available at ScienceDirect

Big Data Research

www.elsevier.com/locate/bdr

SFPM: A Secure and Fine-Grained Privacy-Preserving Matching Protocol for Mobile Social Networking ☆,☆☆

Xue Yang [a,*], Rongxing Lu [b], Hongbin Liang [c], Xiaohu Tang [a]

[a] The Information Security and National Computing Grid Laboratory, Southwest Jiaotong University, Chengdu, 610031, China
[b] School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore
[c] School of Transportation and Logistics, Southwest Jiaotong University, Chengdu, 610031, China

ARTICLE INFO

Article history:
Received 30 May 2015
Received in revised form 9 October 2015
Accepted 3 November 2015
Available online xxxx

Keywords:
Mobile social network
Big data
Proximity-based
Profile matching
Privacy preservation
Fine-grained

ABSTRACT

In emerging big data era, mobile social networking (MSN) is an important data source, which provides an attractive proximity-based communication platform for mobile users with similar interests, attributes, or background to communicate with each other. In this kind of proximity-based MSN, profile matching protocol, which enables a mobile user to break the ice and start a conversation with someone attractive, is one of important components for its success. However, profile matching may occasionally leak the sensitive information, hence privacy concerns often hinder users from enabling this functionality. Aiming at this problem, in this paper, we present a new secure and fine-grained privacy-preserving matching protocol, called SFPM. Differently from those previously reported private profile matching schemes, our proposed SFPM can fine-grainedly differentiate users with the same value of matching metrics by two phases of profile matching. In addition to the personal privacy preservation through secure and efficient cryptographic algorithm, SFPM also achieves the flexibility of profiles changing at the same time. Extensive performance evaluations via smartphones with android system are conducted, and experimental results demonstrate the effectiveness of the SFPM protocol.

1. Introduction

As mentioned by IBM, the rapid development of mobile social networking (MSN) shown in Fig. 1, promotes the generation of big data [1]. Actually, plentiful statistics have indicated that most of big data are produced by MSN, for example, the internet access records of Unicom users have reached 10 TB each day in China. Because of this rising situation, many applications based on big data mining and sharing, like the friend recommender systems of WeChat [2] and Twitter [3], and other personalized recommender systems [4–7], have been emerged. In these applications, when sharing the personal information, like location and preferences in public, people can receive a variety of useful location-based services from these recommender systems. In this paper, we focus on studying a kind of very popular location-based applications, called proximity-based friend recommendation (PFR) mentioned in [8], which allows physically proximate mobile users to have more tangible face-to-face social interaction in public places such as airports, trains and stadiums [9]. In general, one possible way is to use the widely known *profile matching* [10] technique, which is the first step to find the targeting user. As stated by Wu et al. [11], the essence of profile matching is that two users need to compare their personal profile attributes before real interaction. However, a real-world concern is that social profile attributes used in the profile matching process include sensitive information about users and the violation of the privacy of the users' social profiles may pose serious problems. Existing researches show that loss of privacy can expose users to unwanted advertisements [12] and spams/scams, cause social reputation or economic damage [13], and make them victims of blackmail or even physical violence [14]. Hence, the privacy concerns must be addressed when developing profile matching techniques for mobile social networks. In addition to security, clients of mobile social networks run on computing resource-constrained mobile devices. Therefore, a privacy-preserving and power-efficient profile matching scheme is needed for mobile social services.

Recently, there are quite a few schemes for private profile matching, which allow two users to compare their personal profiles without revealing private information to each other [10,15] have been researched. As mentioned in [16], there are two main-

☆ This article belongs to Big Data Networking.
☆☆ Fully documented templates are available in the elsarticle package on CTAN.
* Corresponding author.
E-mail addresses: xueyang.swjtu@gmail.com (X. Yang), rxlu@ntu.edu.sg (R. Lu), hbliang@home.swjtu.edu.cn (H. Liang), xhutang@swjtu.edu.cn (X. Tang).

http://dx.doi.org/10.1016/j.bdr.2015.11.001
2214-5796/

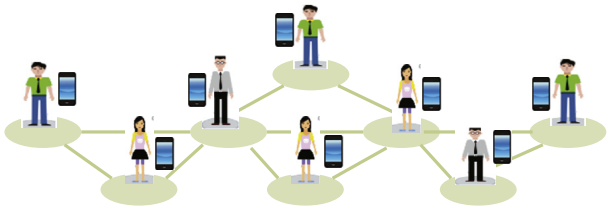


Fig. 1. Popular mobile social networking in big data era.

streams of approaches to solve the privacy-preserving profile-based friend matching problem. The first category treats the personal profile as a set of attributes and provides well-designed protocols based on private set intersection (PSI) and private cardinality of set intersection (PCSI) [10,17,18]; The second category considers the personal profile as a vector and measures the social proximity by private vector dot product or vector distance [19–22]. However, the vast majority of approaches in the first category have been proposed to enable only coarse-grained private matching and are unable to further differentiate users with the same attribute(s), which is less practical in applications [23]. To solve this problem and thus further enhance the usability of PFR in MSN, fine-grained private matching have been widely used in the second category, which are the basic idea of research in this paper. Hence, in what follows, we mainly discuss some related works of the second category.

Liang et al. proposed the multiple pseudonyms technique to improve the anonymity protection for profile matching protocol in [19], where secure dot-product computation is one of important building block. From the perspective of flexibility, multiple pseudonyms technique can ensure anonymity, but, it cannot satisfy the flexibility with slightly larger number of pseudonyms, which actually requires a lot of storage space and management overhead. In [21], Zhang et al. designed a fine-grained private matching protocol with different privacy levels in proximity-based mobile social networks, which included different matching metrics: l_1 distance and max distance. However, it did not consider the difference of profile items and is unable to further differentiate users with the same value of l_1 distance or the max distance. He et al. [24] addressed this issue by proposing a novel user self-controllable profile matching protocol, which allowed users to self-define the weighted of profile items during matching, thus provided more accurate matching results for users. Unfortunately, the method of matching information similarity in both [21] and [24] was based on the time-consuming paillier encryption [25] satisfying homogeneity. Thus, due to the heavy overheads of encryption and decryption, it is difficult to improve the overall operating of MSN applications. The purpose of this paper is to preserve private profile items from disclosing while improving the efficiency of schemes of the second category. In order to improve efficiency, we utilize some efficient methods to securely compute the vector dot product, while existing efficient methods are mainly two kinds. One is a new asymmetric scalar-product-preserving encryption proposed by Wong et al. [22], which is focused on the problem of k-nearest neighbor (kNN) computation on an encrypted database, however, it cannot satisfy the flexibility with the variation of profile items. The other is an efficient privacy-preserving cosine similarity computing (PPCSC) protocol proposed by Lu et al. [26], which could serve as the foundation of many research fields, like privacy-preserving big data miming, data access control, recommendation system. Extensive simulation results showed that the PPCSC protocol is the most efficient one in terms of computation and communication overheads. Thus, we choose the PPCSC protocol as the basis of our protocol. Moreover, most of privacy preserving profile matching protocols do not consider the attack model. To the best of our knowledge, none of the existing solutions to profile matching possesses all the desired properties: privacy-preserving, security (e.g., authentication and integrity), efficiency (e.g., cost-effective computation and communication overhead) and flexibility.

Therefore, how to achieve an efficient, flexible and privacy-preserving profile matching protocol is still challenging in proximity-based MSN. Aiming at the above challenge, in this paper, we propose a secure and finer-grained privacy-preserving matching protocol, called SFPM, for proximity-based MSN. With the SFPM protocol, users can efficiently and flexibly seek out the finer-grained matching target while without disclosing any personal information. In addition, our proposed protocol achieves the integrity of the message and source data authentication, and immensely decreases the computation overhead in comparison with that proposed in [21] and [24], especially alleviating the computational and communication burden of smartphones. Specifically, the main contributions of this paper are four aspects.

- We present SFPM, a new secure and fine-grained privacy-preserving matching protocol, which consists of two stages matching: cosine similarity and weighted l_1 norm. With SFPM, users can finer-grainedly distinguish users and find out the most matched one.
- Compared to the previous private matching protocols, SFPM provides a flexible and efficient matching style. In particular, we introduce a data processing center (DPC) to accomplish matching computations, which can immensely relieve the computation and communication burden of mobile devices. Moreover, the encryption algorithm proposed in [26] is more efficient and flexible compared with [22]. Consider the case when user inserts some profiles, only the inserted profiles should be encrypted, and then DPC only executes multiplication on these profiles and adds them in the previous computation result. Deleting and updating operations are similar with inserting. Therefore, our protocol is flexible for the variation of personal profiles.
- In addition to data confidentiality, the SFPM protocol achieves the integrity of the message and source data authentication by appending the message authentication codes, like the keyed-hashing for message authentication code (HMAC), as a result the ciphertexts can defense the additive noise.
- To validate the effectiveness of the proposed SFPM protocol, we implement both the SFPM protocol and the protocol one proposed by Zhang [21] on a platform with two android phones and a computer. By contrasting, we demonstrate that SFPM is much more efficient than existing similar profile matching schemes [21,24] in terms of the computational overhead.

The remainder of this paper is organized as follows. In Section 2, we formalize the system model and confirm the design goal. After that, we propose the SFPM protocol in Section 3. The security analysis and performance evaluation are introduced in Section 4 and 5, respectively. Finally, we draw our conclusions in Section 6.

2. System model and design goal

2.1. System model

In our system model, we consider a trusted key distribution center (KDC), a semi-trusted data precessing center (DPC), and a group of $l+1$ users $\mathbb{U} = \{U_A, U_1, U_2, \ldots, U_l\}$, where U_A is the requester of the PFR service and the others are the neighbors, as shown in Fig. 2, where we briefly represent the users with smartphones. KDC is a trustable and powerful entity, who is mainly

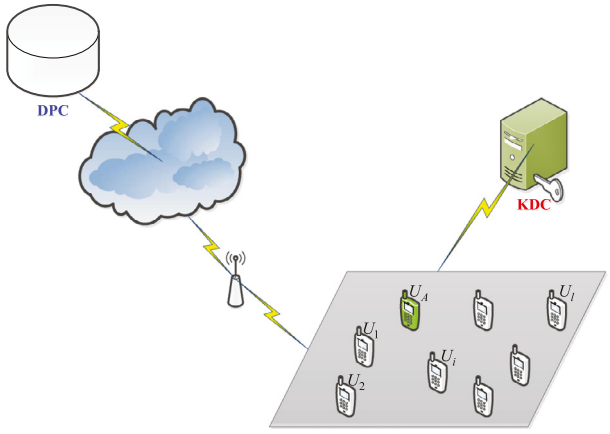


Fig. 2. System model under consideration.

responsible for keys generation of every users. DPC denotes a semi-trusted and powerful entity, who is assumed to be honest-but-curious. That is, it will honestly execute the assigned tasks in the system; however, it would like to learn information of encrypted contents as much as possible. Therefore, it should be prevented from accessing the plaintext even if it is honest. Each user carries a smart-phone or other kinds of mobile devices installing the same matching application, like the friends recommendation of WeChat.

In our SFPM protocol, we essentially define a two-phase profiles matching, which is required to achieve high similarity, efficiency and ensure privacy preserving. Considering a common application scenario, when U_A wants to find a proximity-based person to chat with, he (she) uses the global positioning system (GPS)-based program to find l adjacent users $U_i \in \mathbb{U}$. Applying the two-phase profile matching, U_A may find the most similar user and keep secret of personal profiles simultaneously.

Phase-I matching process. Phase-I matching process is based on the cosine similarity. Once receiving the encrypted profiles, DPC computes the cosine values between U_A and U_i. Let $\mathbb{U}^*$ denote a m-subset of $\mathbb{U}$, which contains m users and anyone of them has the largest cosine value with U_A. If $l \geq m \geq 2$, then m users execute phase-II. Otherwise, the total process is terminated

Phase-II matching process. Phase-II matching process is based on the weighted l_1-norm. DPC computes the weighted l_1-norm between U_A and $U_i \in \mathbb{U}^*$ to find the most similar user, where the weighted vector is according to the profiles of U_A.

2.2. Design goal

Our design goal is to develop a secure and privacy-preserving matching protocol to provide finer-grained profile matching without disclosing any information about their profiles in proximity-based MSN. Specifically, we i) apply an efficient encryption algorithm to minimize the personal profiles disclosure; and ii) develop a two-phase matching method-cosine similarity and weighted l_1-norm to find the most matched user as much as possible.

3. The SFPM protocol

In this section, we propose the SFPM protocol to support high similarity and efficiently profile matching, which consists of three parts: system initialization, cosine similarity profile matching and weighted l_1-norm profile matching. Before describing them, we give two reasonable assumptions: i) the communication channel between KDC and $l+1$ users or DPC are assumed to be secured by existing methods, such as secure sockets layer (SSL); and ii) the communication channels among users are assumed to be 3G, 4G or WiFi networks, which are suitable for those of users and DPC. In addition, for clear description, some notations are used throughout this paper: i) let k_1, k_2, k_3 and k_4 denote the security parameters; ii) $h_{sk}()$ is the HMAC, used for implementing the message integrity and authenticity.

Consider the public attribute set consisting of n attributes $\{A_1, \ldots, A_n\}$, where n may range from several tens to several hundreds depending on specific applications in MSN. The attribute may have different meanings in different contexts, such as interests [10] and disease symptoms [20]. Every user selects an integer $b_i \in [0, q-1]$ to indicate one's level of interest in A_i (for all $i \in [1, n]$). The higher b_i, the more interest the user has in A_i, and vice versa. Therefore, every personal profile is defined as a vector $(b_1, \ldots, b_n) \in \mathbf{F}_q^n$. In particular, we consider U_A with profile $\vec{a} = (a_1, \ldots, a_n)$, and U_j with profile $\vec{b}_j = (b_1^{(j)}, \ldots, b_n^{(j)})$, for $|q| = k_5$, $|n| = k_6$ in the SFPM protocol. When U_A initiates a profile matching request of adjacent people, GPS-based matching applications, like WeChat and Twitter, frequently find the adjacent $U_j \in \mathbb{U}$, who use the same applications. Afterwards, all users, DPC and KDC execute the two-phases profile matching protocol. Because the total privacy-preserving profile matching process between U_A and U_j is identical, for each $1 \leq j \leq l$, without loss of generality, we only consider the profile matching between U_A and U_j, which is shown as in Fig. 3.

3.1. System initialization

KDC first chooses two large primes α, p with $|p| = k_1, |\alpha| = k_2$ and two large random numbers $s, d \in \mathbb{Z}_p$ such that $s \cdot d \equiv 1 \pmod{p}$, and then chooses a secret key K_j ($|K_j| = 128$) for $U_j \in \mathbb{U}$, and K_A ($|K_A| = 128$) for U_A to compute HMAC, respectively. Next, KDC publics the system parameter $params = (p, \alpha)$, and sends (s, K_A) to U_A and (d, K_j) to U_j, respectively. In addition, KDC also sends the key list $L_K = (\langle u_A, K_A\rangle, \langle u_1, K_1\rangle, \ldots, \langle u_l, K_l\rangle)$ to DPC, where u_A and u_j are the unique identities for U_A and U_j, respectively.

3.2. Phase-I: cosine similarity matching

Before describing the detailed process, we give the well-known formula called cosine similarity.

$$\cos(\vec{u}, \vec{v}) = \frac{\sum_{i=1}^{n} u_i v_i}{\sqrt{\sum_{i=1}^{n} u_i^2}\sqrt{\sum_{i=1}^{n} v_i^2}}.$$

This is an important measure of similarity of two objectives captured by vectors $\vec{u} = (u_1, \ldots, u_n)$ and $\vec{v} = (v_1, \ldots, v_n)$, respectively. In big data analysis, $\cos(\vec{u}, \vec{v})$ has become a critical building block for many data mining techniques.

Based on the cosine similarity formula, we can obtain that the greater $\cos(\vec{a}, \vec{b}_j)$ of U_A and U_j is, the higher the cosine similarity, which implies the higher matching degree between U_A and U_j. The detailed operations of phase-I are as follows.

1. U_A executes the following operations:
   a For each $a_i, i = 1, \ldots, n$, choose a random numbers c_i, where c_i is of the length $|c_i| = k_3$ bits, and compute
   $$C_i = s(\alpha a_i + c_i) \pmod{p}. \quad (1)$$
   In addition, compute $A = \sum_{i=1}^{n} a_i^2$.
   b Compute the HMAC for $u_A \| A$ by means of the key K_A, e.g. $h_{K_A}(u_A \| A)$.
   c Send $(u_A, A, C_1, \ldots, C_n, h_{K_A}(u_A \| A))$ to DPC.
2. U_j does the following operations:

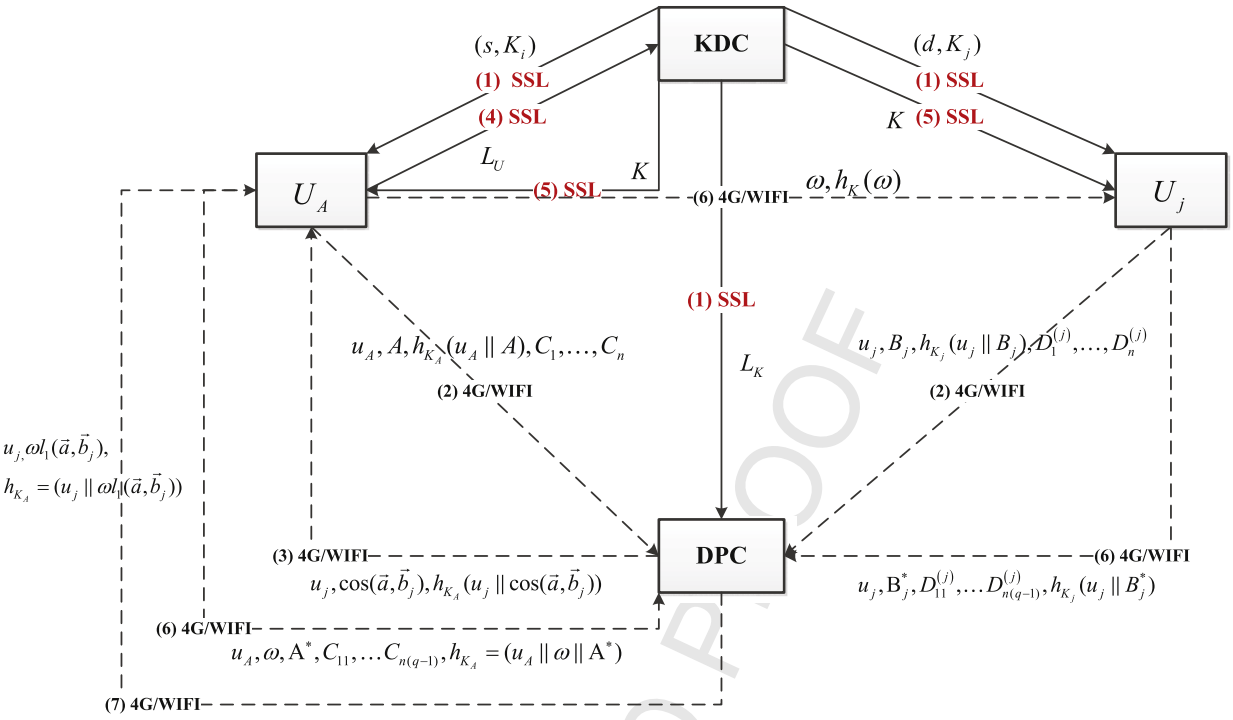


Fig. 3. The SFPM protocol of fine-grained privacy-preserving matching.

a For each $b_i^{(j)}, i = 1, \ldots, n$, choose a random numbers $r_i^{(j)}$, where $r_i^{(j)}$ is of the length $|r_i^{(j)}| = k_4$ bits, and compute

$$D_i^{(j)} = d(\alpha b_i^{(j)} + r_i^{(j)}) \pmod{p}. \quad (2)$$

Besides, compute $B_j = \sum_{i=1}^{n} (b_i^{(j)})^2$.

b Compute the HMAC for $u_j \| B_j$ by means of the key K_j, e.g. $h_{K_j}(u_j \| B_j)$.

c Send $(u_j, B_j, D_1^{(j)}, \ldots, D_n^{(j)}, h_{K_j}(u_j \| B_j))$ to DPC.

3. DPC does the following after receiving the messages from U_A and U_j.

a Refer to the key list L_K stored in advance, and find the HMAC key K_A and K_j of U_A and U_j, respectively. With the HMAC keys, DPC computes $h'_{K_A}(u_A \| A)$ and $h'_{K_j}(u_j \| B_j)$, respectively. Afterwards, check whether $h'_{K_A}(u_A \| A) = h_{K_A}(u_A \| A)$ and $h'_{K_j}(u_j \| B_j) = h_{K_j}(u_j \| B_j)$.

b If the verification fails, DPC requires retransmission or does nothing. Otherwise, DPC does the following operations:

1) Compute

$$E = \sum_{i=1}^{n} C_i D_i^{(j)} \pmod{p}. \quad (3)$$

And then, obtain the dot product $\vec{a} \cdot \vec{b}_j = \sum_{i=1}^{n} a_i b_i^{(j)} = \frac{E - (E \pmod{\alpha^2})}{\alpha^2}$, so $\cos(\vec{a}, \vec{b}_j) = \frac{\vec{a} \cdot \vec{b}_j}{\sqrt{A} \cdot \sqrt{B_j}}$.

2) After l cosine values have been computed, denoted as $\{u_j, \cos(\vec{a}, \vec{b}_j)\}_{j=1}^{l}$. DPC computes the HMAC $h_{K_A}(\{u_j \| \cos(\vec{a}, \vec{b}_j)\}_{j=1}^{l})$ by means of the secret key K_A.

c Return $(\{u_j, \cos(\vec{a}, \vec{b}_j)\}_{j=1}^{l}, h_{K_A}(\{u_j \| \cos(\vec{a}, \vec{b}_j)\}_{j=1}^{l}))$ back to U_A

4. Once receiving the message, U_A computes $h'_{K_A}(\{u_j \| \cos(\vec{a}, \vec{b}_j)\}_{j=1}^{l})$ and checks whether $h'_{K_A}(\{u_j \| \cos(\vec{a}, \vec{b}_j)\}_{j=1}^{l}) = h_{K_A}(\{u_j \| \cos(\vec{a}, \vec{b}_j)\}_{j=1}^{l})$. If the verification fails, U_A requires retransmission. Otherwise, he (she) compares l values of $\cos(\vec{a}, \vec{b}_j)$, where $j = 1, \ldots, l$, and then chooses a set of m users $\mathbb{U}^*$, where $\mathbb{U}^* = \{U_i | i = \arg\max_{j=1,\ldots,l} \cos(\vec{a}, \vec{b}_j)\}$.

After phase-I is finished, m remaining users belong to the subset $\mathbb{U}^*$, where $0 \le m \le l$. If $m < 2$, the total process is terminated, otherwise, for each $U_j \in \mathbb{U}^*$, DPC executes the following weighted l_1-norm matching between U_A and U_j.

3.3. Phase-II: weighted l_1-norm matching

Based on the result of phase-I, m users remain to be matched with U_A to seek out the most matched one. Before describing the detailed operations, we give a definition of the binary vector.

Definition 1. For all $x \in [0, q-1]$, we define a binary vector $h(x) = (x_1, \ldots, x_{q-1})$, where x_i is equal to one for $1 \le i \le x$ and zero for $x < i \le q-1$.

Recall that the profiles of U_A is $\vec{a} = (a_1, \ldots, a_n)$ and U_j is $\vec{b}_j = (b_1^{(j)}, \ldots, b_n^{(j)})$. Therefore, the l_1-norm between $\vec{a}$ and $\vec{b}_j$ is defined as

$$l_1(\vec{a}, \vec{b}_j) = \sum_{i=1}^{n} |a_i - b_i^{(j)}|.$$

The l_1-norm evaluates the overall absolute difference between two personal profiles. However, we can not compute l_1-norm immediately so that the privacy of personal profiles can be prevented from disclosing. In particular, by means of Definition 1, we convert $\vec{a}$, $\vec{b}_j$ into

$$\hat{a} = h(\vec{a}) = (\hat{a}_{11}, \ldots, \hat{a}_{1(q-1)}, \ldots, \hat{a}_{ik}, \ldots, \hat{a}_{n(q-1)})$$
$$\hat{b}_j = h(\vec{b}_j) = (\hat{b}_{11}^{(j)}, \ldots, \hat{b}_{1(q-1)}^{(j)}, \ldots, \hat{b}_{ik}^{(j)}, \ldots, \hat{b}_{n(q-1)}^{(j)}),$$

where $h(\vec{a}) = (h(a_1), \ldots, h(a_n))$, $h(\vec{b}_j) = (h(b_1^{(j)}), \ldots, h(b_n^{(j)}))$, $i = 1, \ldots, n$ and $k = 1, \ldots, q-1$. Therefore, with the appropriate

weight vector $\omega = (\omega_1, \ldots, \omega_n)$ chosen by U_A, the weighted l_1-norm can be derived as follows:

$$\begin{aligned}
\omega l_1(\vec{a}, \vec{b}_j) &= \sum_{i=1}^{n} \omega_i |a_i - b_i^{(j)}| \\
&= \sum_{i=1}^{n} \omega_i \Big(\sum_{k=1}^{q-1} |\hat{a}_{ik} - \hat{b}_{ik}^{(j)}|\Big) \\
&\xrightarrow{\because \hat{a}_{ik}, \hat{b}_{ik}^{(j)} \in \{0,1\}} \\
&= \sum_{i=1}^{n} \omega_i \Big(\sum_{k=1}^{q-1} |\hat{a}_{ik} - \hat{b}_{ik}^{(j)}|^2\Big) \\
&= \sum_{i=1}^{n} \omega_i \Big(\sum_{k=1}^{q-1} (\hat{a}_{ik}^2 + (\hat{b}_{ik}^{(j)})^2 - 2\hat{a}_{ik} \cdot \hat{b}_{ik}^{(j)})\Big) \\
&= \sum_{i=1}^{n} \omega_i a_i + \sum_{i=1}^{n} \omega_i b_i^{(j)} - 2\sum_{i=1}^{n} \omega_i \Big(\sum_{k=1}^{q-1} \hat{a}_{ik} \cdot \hat{b}_{ik}^{(j)}\Big) \\
&= \sum_{i=1}^{n} \omega_i a_i + \sum_{i=1}^{n} \omega_i b_i^{(j)} - 2\sum_{i=1}^{n} \omega_i h(a_i) h(b_i^{(j)}).
\end{aligned}$$

When the third equality holds because the value of $\hat{a}_{ik}$ and $\hat{b}_{ik}^{(j)}$ is either one or zero, that is, the value of $|\hat{a}_{ik} - \hat{b}_{ik}^{(j)}|$ is either one or zero.

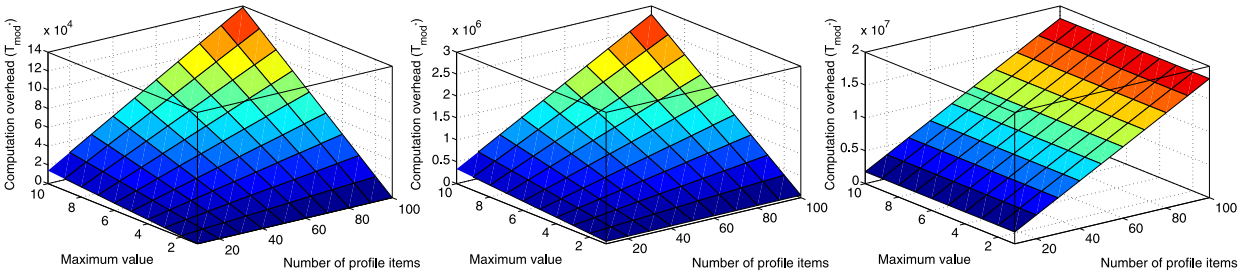
Since U_A and U_j possess the values $\sum_{i=1}^{n} \omega_i a_i$ and $\sum_{i=1}^{n} \omega_i b_i^{(j)}$, respectively, we only need to securely compute the value $h(a_i) \cdot h(b_i^{(j)})$ for $i = 1, \ldots, n$, without knowing profiles of U_A and U_j. As in phase-I, U_A, DPC and each $U_j \in \mathbb{U}^*$ execute the following operations.

1. U_A sends the identities list L_U of m users to KDC, e.g. $L_U = (u_1, \ldots, u_m)$.
2. After receiving L_U, KDC chooses the secret key K with $|K| = 128$, to compute HMAC, and then sends it to U_j and U_A, respectively.
3. Once receiving the secret key K, U_A first chooses the weight vector $\omega = (\omega_1, \ldots, \omega_n)$ based on the preference of profiles $\vec{a}$, and computes the HMAC for ω, e.g. $h_K(\omega)$. And then implements the following operations:
   a Send the message $(\omega, h_K(\omega))$ to U_j.
   b Convert the profile vector $\vec{a} = (a_1, \ldots, a_n)$ into $\hat{a} = (\hat{a}_{11}, \ldots, \hat{a}_{n(q-1)})$. For each $\hat{a}_{ik} \in \hat{a}$, choose a random numbers c_{ik}, where c_{ik} is of the length $|c_{ik}| = k_3$ bits, and calculate
   $$C_{ik} = s(\alpha \hat{a}_{ik} + c_{ik}) \pmod{p}. \quad (4)$$
   Besides, compute $A^* = \sum_{i=1}^{n} \omega_i a_i$
   c Compute the HMAC for $u_A \| A^* \| \omega$ with the key K_A, e.g. $h_{K_A}(u_A \| A^* \| \omega)$.
   d Send the message $(u_A, A^*, \omega, C_{11}, \ldots, C_{n(q-1)}, h_{K_A}(u_A \| A^* \| \omega))$ to DPC.
4. Once receiving the $(\omega, h_K(\omega))$, U_j checks the validity of HMAC. If verification succeeds, U_j executes the following operations:
   a Convert profile vector $\vec{b}_j = (b_1^{(j)}, \ldots, b_n^{(j)})$ into $\hat{b}_j = (\hat{b}_{11}^{(j)}, \ldots, \hat{b}_{n(q-1)}^{(j)})$. For each $\hat{b}_{ik}^{(j)} \in \hat{b}_j$, choose a random numbers $r_{ik}^{(j)}$, where $r_{ik}^{(j)}$ is of the length $|r_{ik}^{(j)}| = k_4$ bits, and compute
   $$D_{ik}^{(j)} = d(\alpha \hat{b}_{ik}^{(j)} + r_{ik}^{(j)}) \pmod{p}. \quad (5)$$
   In addition, compute $B_j^* = \sum_{i=1}^{n} \omega_i b_i^{(j)}$.
   b Compute the HMAC for $u_j \| B_j^*$ by means of the key K_j, e.g. $h_{K_j}(u_j \| B_j^*)$.
   c Send $(u_j, B_j^*, D_{11}^{(j)}, \ldots, D_{n(q-1)}^{(j)}, h_{K_j}(u_j \| B_j^*))$ to DPC.
5. After receiving the messages from U_A and U_j, DPC checks the validity of HMACs, e.g. $h_{K_j}(u_j \| B_j^*)$ and $h_{K_A}(u_A \| A^* \| \omega)$. If verification succeeds, DPC executes the following operations:
   a For each $1 \le i \le n$, compute
   $$E_i = \sum_{k=1}^{q-1} C_{ik} D_{ik}^{(j)} \pmod{p}. \quad (6)$$
   And then, compute $\omega_i h(a_i) h(b_i^{(j)}) = \omega_i \frac{E_i - (E_i \pmod{\alpha^2})}{\alpha^2}$. Consequently, the weighted l_1-norm between U_A and U_j e.g. $\omega l_1(\vec{a}, \vec{b}_j) = A^* + B_j^* - 2\sum_{i=1}^{n} \omega_i h(a_i) h(b_i^{(j)})$, can be calculated.
   b After m values of the weighted l_1-norm have been computed, denoted as $\{u_j, \omega l_1(\vec{a}, \vec{b}_j)\}_{j=1}^{m}$, compute the HMAC $h_{K_A}(\{u_j \| \omega l_1(\vec{a}, \vec{b}_j)\}_{j=1}^{m})$ by means of the secret key K_A, and return
   $$(\{u_j, \omega l_1(\vec{a}, \vec{b}_j)\}_{j=1}^{m}, h_{K_A}(\{u_j \| \omega l_1(\vec{a}, \vec{b}_j)\}_{j=1}^{m}))$$
   back to U_A.
6. After receiving the message from DPC, U_A first checks the validity of the HMAC. If the verification fails, U_A requires retransmission. Otherwise, he (she) compares m values of $\omega l_1(\vec{a}, \vec{b}_j)$, where $j = 1, \ldots, m$, and find out the most matched one U_B, where $U_B \in \mathbb{U}^*$, such that $B = \arg\min_{j=1,\ldots,m} \omega l_1(\vec{a}, \vec{b}_j)$.

4. Security analysis

In this section, we analyze the security and feasibility of the SFPM protocol. As discussed in Section 1, each $U_j \in \mathbb{U}$ and U_A concern about disclosing their personal profiles to complete strangers, so a privacy-preserving matching protocol is needed. For the privacy preservation of the SFPM protocol, since each $a_i \in \vec{a}$ or $\hat{a}_{ik} \in \hat{a}$ is one time masked with random $C_i = s(\alpha a_i + c_i) \pmod{p}$ or $C_{ik} = s(\alpha \hat{a}_{ik} + c_{ik}) \pmod{p}$, respectively, without knowing the private key s and the random numbers c_i or c_{ik}, it is impossible to guess U_A's vector $\vec{a} = (a_1, \ldots, a_n)$. In addition, for ensuring the confidentiality of the private key s, the parameter p is set large enough, which also increases the difficulty for guessing the vector $\vec{a}$. Therefore, each $a_i \in \vec{a}$ is privacy-preserving during the dot-product computation. For each $U_j \in \mathbb{U}$, the security analysis for the profile vector $\vec{b}_j = (b_1^{(j)}, \ldots, b_n^{(j)})$ is similar with that of U_A. Hence, the profiles of U_j are also privacy-preserving during the dot-product computation. Because the communication channels between DPC and users or among users are assumed to 3G, 4G or WiFi networks, which are insecure, the message transmitted through these channels would be likely to suffer from some active attacks, like message tampering. Hence, we utilize HMACs in our proposed SFPM protocol, to ensure the integrity of the message and source data authentication. In addition, the ciphertexts can defense the additive noise. For example, in phase-I, if the ciphertexts of U_A is tampered by adding the additive noise denoted as N, the Eq. (1) becomes $C_i' = s(\alpha a_i + c_i) + N \pmod{p}$, then DPC computes

$$\begin{aligned}
E' &= \sum_{i=1}^{n} C_i' D_i^{(j)} \\
&= \sum_{i=1}^{n} [\alpha^2 a_i b_i^{(j)} + \alpha a_i r_i^{(j)} + \alpha c_i b_i^{(j)} + c_i r_i^{(j)}
\end{aligned}$$



(a) SFPM protocol (b) Protocol one [21] (c) He *et al.*'s protocol [24]

Fig. 4. Comparison of computation complexity.

$$+ N(\alpha b_i^{(j)} + r_i^{(j)})] \pmod p,$$

and then, computes $E' - (E' \pmod{\alpha^2}) = \alpha^2 a_i b_i^{(j)}$, which equals to $E - (E \pmod{\alpha^2})$, that is the ciphertexts can resist the additive noise. Obviously, the ciphertexts of U_j also can resist the addictive noise.

Finally we give restrictions of system parameters for the feasibility of the SFPM protocol. In fact, the key lies in correctly obtaining

$$\sum_{i=1}^{n} a_i b_i^{(j)} = \frac{E - (E \pmod{\alpha^2})}{\alpha^2}. \tag{7}$$

Hence, we need constraints such that

$$\sum_{i=1}^{n} (\alpha^2 a_i b_i^{(j)} + \alpha a_i r_i^{(j)} + \alpha b_i^{(j)} c_i + c_i r_i^{(j)}) < p \tag{8}$$

$$\sum_{i=1}^{n} (\alpha a_i r_i^{(j)} + \alpha b_i^{(j)} c_i + c_i r_i^{(j)}) < \alpha^2. \tag{9}$$

5. Performance evaluation

In this section, we evaluate the computation complexity and communication overhead as well as overall execution time of our protocol in contrast to two similar works proposed by Zhang et al. (Protocol one in [21]) and He et al. [24], respectively. As mentioned in [24], since these two protocols do not consider the Phase-I matching process (the cosine similarity), we only need to compare the computation and communication overhead in our Phase-II matching parts with those in their protocols. In addition, for convenience of comparison, we only consider the matching execution between two users, e.g., U_A and U_j.

5.1. Computation complexity

From the proposed SFPM scheme, when two users execute the Phase-II matching, the computation overhead incurred is mainly related to modular multiplication, modular addition and modular arithmetic. In particular, U_A needs to perform $n(q-1)$ encryptions executed by smartphones, each costing two 1024-bit multiplications (mul_1) and one 1024-bit addition (add) according to Eq. (4). Similarly, U_j totally needs $2n(q-1)$ mul_1 and $n(q-1)$ add. Hence, the total computation overhead for two users are $4n(q-1)mul_1 + 2n(q-1)add$. As for DPC, it needs perform n weighted l_1 norm matching, each costing $(q-1)$ 1024-bit multiplications (mul_1^*), $(q-2)$ 1024-bit additions (add^*) and one $2k_2$-bit modular arithmetic (mod^*) according to Eq. (6). Hence, we can have that the total computation overhead of SFPM protocol is $(4n(q-1)\ mul_1 + 2n(q-1)\ add + n(q-1)\ mul_1^* + n(q-2)\ add^* + n\ mod^*)$ approximately.

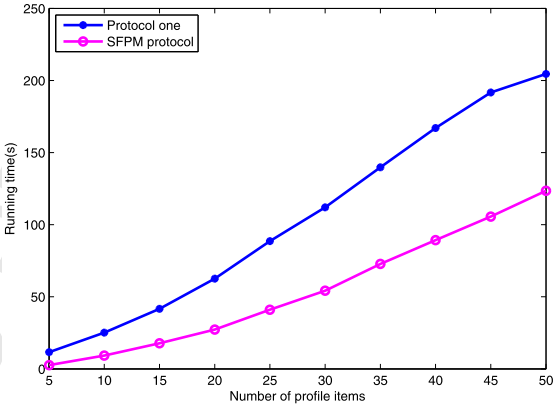
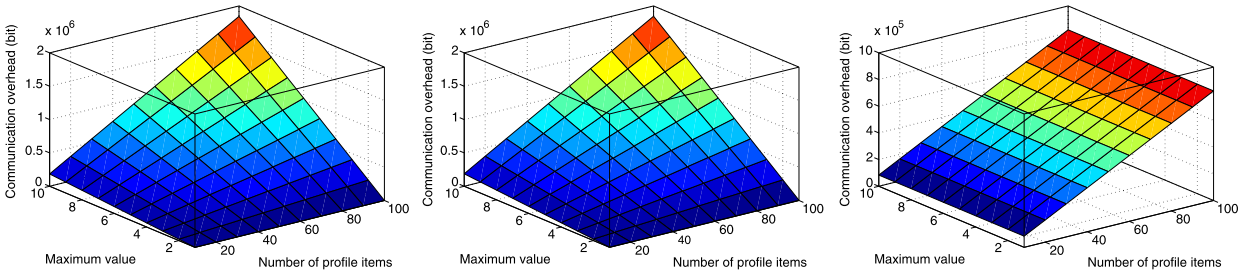


Fig. 5. Average running time vs. number of profile items.

Table 1
Comparison of computation complexity.

	Computation
SFPM	$4n(q-1)\ mul_1 + 2n(q-1)\ add$ $+ n(q-1)\ mul_1^* + n(q-2)\ add^* + n\ mod^*$
Protocol one [21]	$2\ exp_2 + (2qn - 2n + 2)\ exp_1$ $+ (qn - n + \sum_{i=1}^{n} b_i + 1) mul_2$
He et al.'s protocol [24]	$(12\ exp_1 + 6\ exp_2 + 13\ mul_2)n$

We present the computation complexity comparison of SFPM, the protocol one in [21] and He et al.'s protocol in [24] in Table 1, where mul_2, exp_1 and exp_2 denote one 2048-bit multiplication, 1024-bit exponentiation and 2048-bit exponentiation executed by users (smartphones) in [21] and [24], respectively. Furthermore, we simulate operations on two MeiZu phones with android system 4.1 and 16 GB of RAM for evaluation. In addition, we use a computer with an Intel(R) Core(TM) i5-4460T CPU running at 1.90 GHz, and with 8 GB of RAM, which is acted as DPC. According to the benchmark test results in [27], the experimental result indicate that $T_{add^*} = T_{mod^*}$, $T_{add} = 11T_{mod^*}$, $T_{mul_1^*} = 3T_{mod^*}$, $T_{mul_1} = 22T_{mod^*}$, $T_{mul_2} = 32T_{mod^*}$, $T_{exp_1} = 1513T_{mod^*}$ and $T_{exp_2} = 27\,080T_{mod^*}$. From Table 1, we can see that besides varying with n and q, the computation overhead of the protocol one is also influenced by the real value of b_i, where $b_i \in [0, q-1]$ and $i = 1, \ldots, n$. In order to simulate the comparison of the theoretical analysis based on Table 1, we consider the best case $b_i = 0, i = 1, \ldots, n$ for the protocol one, that is the computation overhead of the protocol one is the least. Then, with the exact operation costs, we depict the variation of computation costs in terms of number of profile n and maximum value of profile items q in Fig. 4. From the figure, it is obviously shown that the SFPM scheme largely reduces the computation complexity compared to the protocol one [21] and He et al.'s protocol [24]. It is worth noting that the number of pro-



(a) SFPM protocol (b) Protocol one [21] (c) He *et al.*'s protocol [24]

Fig. 6. Comparison of communication overhead.

Table 2
Comparison of communication overhead.

	SFPM	Protocol one [21]	He et al.'s protocol [24]
Comm.	$2048n(q-1)+480$	$2048n(q-1)+3232$	$2048(4n+1)$

file n is varying from 10 to 100 and maximum value of profile items q is varying from 1 to 10, which are sufficient to specific MSN applications. Because, as mentioned in [21], the number of profile n may range from several tens to several hundreds and the maximum value of profile items q could be a small integer, say 5 or 10, which may be sufficient to differentiate user's interest level.

Moreover, we implement our protocol and the compared protocol one [21] between two users with the number n of attributes varying. To facilitate the comparison, according to inequality (8) and (9), if we choose the length of p to be 1024-bits, we can just set $k_2 = 200$, $k_3 = k_4 = 128$, which can ensure that we get the correct result. Specifically, in order to quickly get the results of simulations, we just set $q = 32$. For each parameter setting, we run experiments 50 times, and obtain the average total computation overhead. As shown in Fig. 5, the gap of computation overhead is gradually increasing with varying n, and our protocol is more efficient than the protocol one in terms of the computation complexity.

5.2. Communication overhead

The communication overhead incurred by the SFPM protocol involves two aspects: 1) U_A and U_j send ciphertexts and HMACs to DPC. The data is in the form of $(u_j, B_j^*, D_{11}^{(j)}, \ldots, D_{n(q-1)}^{(j)}, h_{K_j}(u_j \| B_j^*))$ for U_j and its size should be $Su = 1024n(q-1) + |u_j| + |B_j^*| + 128$ if we choose the length of p and HMAC to be 1024-bits and 128-bits respectively; 2) DPC sends the result to U_A. The data is in the form of $(\{u_j, \omega l_1(\vec{a}, \vec{b}_j)\}_{j=1}^m, h_{K_A}(\{u_j \| \omega l_1(\vec{a}, \vec{b}_j)\}_{j=1}^m))$ and its size should be $Sd = |\{u_j, \omega l_1(\vec{a}, \vec{b}_j)\}_{j=1}^m| + 128$. Similarly as the data processing method in [28], we set $|u_j| + |B_j^*| = |\{u_j, \omega l_1(\vec{a}, \vec{b}_j)\}_{j=1}^m|$ as 32-bit length. Hence, we can approximate the total communication overhead to be $2048n(q-1)+480$ bits.

We present the communication overhead comparison of SFPM, the protocol one in [21] and He et al.'s protocol in [24] in Table 2. Furthermore, we plot the overall communication overhead of three protocols with respect to the variance of n and q in Fig. 6. From the figures we can see that when $q \leq 5$, the communication overhead of our protocol is always less than other two protocols with varying n. If $q > 6$, the communication overhead of our protocol is little larger than those of He et al.'s protocol [24], but is always less than the protocol one in [21]. As shown in Fig. 3, the main reason are two aspects: i) for ensuring the integrity of message and source data authentication, we introduce the HMACs, so each message transmitted among users and DPC should include the HMACs; ii) the identity of each $U_j \in \mathbb{U}$ is transmitted for differentiating users. Actually, with the rapidly mobile telecommunication technology, known as 4G or 5G, it will allow larger bandwidth. For example, existing 4G technology can offer efficient and quite sufficient communication bandwidth for PFR applications. Accordingly, in order to resist some active attacks (e.g., message tampering and forgery attack), although our protocol increases a certain amount of communication overhead, like HMAC, this does not influence the efficiency of current communication.

6. Conclusion

In this paper, we have proposed a secure and fine-grained privacy-preserving matching protocol for mobile social networking (MSN), which provides the fundamental step of effective communication among strangers and prevents personal privacy from disclosing simultaneously. In addition, the SFPM protocol realizes the finer-grained differentiation of personal profiles and the flexibility of the cryptographic algorithm. Detailed security analysis shows that the proposed SFPM protocol can ensure privacy preserving, integrity of the communication message and source data authentication. In addition, the additive noise can be resisted in our protocol. Finally, the performance evaluation implemented on a platform with two android phones and a computer verifies the effectiveness of the proposed SFPM protocol. For the convenience of comparison between the proposed SFPM protocol and the protocol one, we just utilize the l_1 norm to measure the similarity between two users, however, there are many other metrics for evaluating the similarity. Therefore, our future work is to utilize the other metrics, like l_2 norm, to further promote the efficiency of the total protocol. In addition, we will implement the proposed SFPM protocol in other application environments.

Appendix A. Supplementary material

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.bdr.2015.11.001.

References

[1] IBM, Big data at the speed of business, http://www-01.ibm.com/.
[2] J. Bo, Q. Yanmei, Design and implementation of mobile library APP service system based on WeChat, J. Mod. Inf. 33 (6) (2013) 41–44.
[3] A. Tumasjan, T.O. Sprenger, P.G. Sandner, I.M. Welpe, Predicting elections with twitter: what 140 characters reveal about political sentiment, in: ICWSM, vol. 10, 2010, pp. 178–185.
[4] W. Woerndl, C. Schueller, R. Wojtech, A hybrid recommender system for context-aware recommendations of mobile applications, in: 2007 IEEE 23rd International Conference on Data Engineering Workshop, IEEE, 2007, pp. 871–878.
[5] Y. Zheng, Y. Chen, X. Xie, W.-Y. Ma, Geolife2. 0: a location-based social networking service, in: Tenth International Conference on Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09, IEEE, 2009, pp. 357–358.
[6] F. Ricci, L. Rokach, B. Shapira, Introduction to recommender systems handbook, in: Recommender Systems Handbook, Springer, 2011, pp. 1–35.

[7] S. Shirwadkar, S. Yami, Method and system for searching location based information on a mobile device, vol. 12, Feb. 2004, US Patent App. 10/777,237.
[8] D. Quercia, L. Capra, Friendsensing: recommending friends using mobile phones, in: Proceedings of the Third ACM Conference on Recommender Systems, ACM, 2009, pp. 273–276.
[9] Z. Yang, B. Zhang, J. Dai, A.C. Champion, D. Xuan, D. Li, E-smalltalker: a distributed mobile system for social networking in physical proximity, in: 2010 IEEE 30th International Conference on Distributed Computing Systems (ICDCS), IEEE, 2010, pp. 468–477.
[10] M. Li, N. Cao, S. Yu, W. Lou, Findu: privacy-preserving personal profile matching in mobile social networks, in: INFOCOM, 2011 Proceedings IEEE, IEEE, 2011, pp. 2435–2443.
[11] X. Wu, X. Zhu, G.-Q. Wu, W. Ding, Data mining with big data, IEEE Trans. Knowl. Data Eng. 26 (1) (2014) 97–107.
[12] R.A. Popa, A.J. Blumberg, H. Balakrishnan, F.H. Li, Privacy and accountability for location-based aggregate statistics, in: Proceedings of the 18th ACM Conference on Computer and Communications Security, ACM, 2011, pp. 653–666.
[13] C. Kaufman, R. Perlman, M. Speciner, Network Security: Private Communication in a Public World, Prentice Hall Press, 2002.
[14] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, J.-Y. Le Boudec, Protecting location privacy: optimal strategy against localization attacks, in: Proceedings of the 2012 ACM Conference on Computer and Communications Security, ACM, 2012, pp. 617–627.
[15] W. Dong, V. Dave, L. Qiu, Y. Zhang, Secure friend discovery in mobile social networks, in: INFOCOM, 2011 Proceedings IEEE, IEEE, 2011, pp. 1647–1655.
[16] Y. Wang, J. Xu, Overview on privacy-preserving profile-matching mechanisms in mobile social networks in proximity (msnp), in: 2014 Ninth Asia Joint Conference on Information Security (ASIA JCIS), IEEE, 2014, pp. 133–140.
[17] E. De Cristofaro, G. Tsudik, Practical private set intersection protocols with linear complexity, in: Financial Cryptography and Data Security, Springer, 2010, pp. 143–159.
[18] G. Costantino, F. Martinelli, P. Santi, Privacy-preserving mobility-casting in opportunistic networks, in: 2014 Twelfth Annual International Conference on Privacy, Security and Trust (PST), IEEE, 2014, pp. 10–18.
[19] X. Liang, R. Lu, X. Lin, X.S. Shen, Profile matching protocol with anonymity enhancing techniques, in: Security and Privacy in Mobile Social Networks, Springer, 2013, pp. 19–41.
[20] R. Lu, X. Lin, X. Shen, SPOC: a secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency, IEEE Trans. Parallel Distrib. Syst. 24 (3) (2013) 614–624.
[21] R. Zhang, Y. Zhang, J. Sun, G. Yan, Fine-grained private matching for proximity-based mobile social networking, in: INFOCOM, 2012 Proceedings IEEE, IEEE, 2012, pp. 1969–1977.
[22] W.K. Wong, D.W.-l. Cheung, B. Kao, N. Mamoulis, Secure kNN computation on encrypted databases, in: Proceedings of the 2009 ACM SIGMOD International Conference on Management of Data, ACM, 2009, pp. 139–152.
[23] H. Zhu, S. Du, M. Li, Z. Gao, Fairness-aware and privacy-preserving friend matching protocol in mobile social networks, IEEE Trans. Emerg. Top. Comput. 1 (1) (2013) 192–200.
[24] D. He, Z. Cao, X. Dong, J. Shen, User self-controllable profile matching for privacy-preserving mobile social networks, in: 2014 IEEE International Conference on Communication Systems (ICCS), IEEE, 2014, pp. 248–252.
[25] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: Advances in Cryptology—EUROCRYPT'99, Springer, 1999, pp. 223–238.
[26] R. Lu, H. Zhu, X. Liu, J.K. Liu, J. Shao, Toward efficient and privacy-preserving computing in big data era, IEEE Netw. 28 (4) (2014) 46–50.
[27] S. Bhatt, R. Sion, B. Carbunar, A personal mobile DRM manager for smartphones, Comput. Secur. 28 (6) (2009) 327–340.
[28] R. Lu, X. Liang, X. Li, X. Lin, X.S. Shen, EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications, IEEE Trans. Parallel Distrib. Syst. 23 (9) (2012) 1621–1631.

Please cite this article in press as: X. Yang et al., SFPM: A Secure and Fine-Grained Privacy-Preserving Matching Protocol for Mobile Social Networking, Big Data Research (2015), http://dx.doi.org/10.1016/j.bdr.2015.11.001