NMHP: A Privacy Preserving Profile Matching Protocol in Multi-hop Proximity Mobile Social Networks

Entao Luo[1,3], Qin Liu[2], and Guojun Wang[1,4](✉)

[1] School of Information Science and Engineering, Central South University,
Changsha 410083, China
{cs_entaoluo,csgjwang}@csu.edu.cn

[2] School of Information Science and Engineering, Hunan University,
Changsha 410082, China
gracelq628@hnu.edu.cn

[3] School of Electronics and Information Engineering,
Hunan University of Science and Engineering, Yongzhou 425199, China

[4] School of Computer Science and Educational Software,
Guangzhou University, Guangzhou 510006, China

Abstract. With the rapid development of mobile devices and online social networks, users in Proximity-based Mobile Social Networks (PMSNs) can easily discover and make new social interactions with others by profile matching. The profiles usually contain sensitive personal information, while the emerging requirement of profile matching in proximity mobile social networks may occasionally leak the sensitive information and hence violate people's privacy. In this paper, we propose a multi-hop profile matching protocol (NMHP) in PMSNs. By using our protocol, users can customize the matching matrices to involve their own matching preference and to make the matching results more precise. In addition, to achieve a secure and efficient matching, we utilize the confusion matrix transformation and the idea of multi-hop, which means we make profile matching within several hops instead one. Security analysis shows that our proposed protocol can realize privacy-preserving friend discovery with higher efficiency.

Keywords: Profile matching · Friend discovery · Trusted third party · Confusion matrix · Dot production

1 Introduction

With the increasing popularity of mobile devices (e.g., smart phones) and the great advancement of online social networking, Mobile Social Networks (MSN) have become a vital part in our daily life [20]. Especially, the explosive growth of mobile-connected and location-aware devices makes it possible and meaningful to do MSN in proximity (PMSNs) [16,17]. Proximity-based Mobile Social Networks

G. Wang et al. (Eds.): ICA3PP 2015, Part III, LNCS 9530, pp. 463–474, 2015.
DOI: 10.1007/978-3-319-27137-8_34

(PMSNs) is one of the fastest-growing activities among mobile users. Nowadays, users can discover and make new social interactions easily with physical-proximate mobile users through Wi-Fi or Bluetooth interfaces embedded in their Smartphone or Tablet.

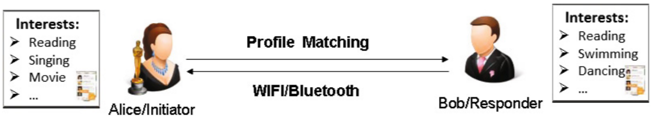

Fig. 1. Profile matching in friend discovery

In PMSNs, in order to join and enjoy the social activities, users usually begin by creating a profile, then interact with other users. In the process, it is necessary to disclose some personal and private information, it is clearly that, the user's privacy may be revealed during such process. However, users' profile may include some sensitive information, so it is dangerous to leak them to nearby strangers.

In order to solve this problem, a group of private matching protocols have been proposed recently, among which these protocols, some protect users' privacy with reliance on a TTP (Trusted Third Party) [3,8,10,12]. In TTP-based approaches, 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.

In the other TTP-free schemes, there are two mainstreams of approaches to solve this problem. The first category treats a user's profile as a set of attributes and provides private attributes matching based on private set intersection (PSI) and private cardinality of set intersection (PCSI) [7,15]. The second category considers a user's profile as a vector and measures the social proximity by private vector dot production [5,19].

Although the above mentioned protocols can provide private matching, most of them use complicated cryptographic computation to ensure the privacy, which are not efficient enough for the resource-restricted mobile devices. For example, homomorphic encryption and decryption are used in [4,6,13], introducing more computation overhead due to modulus exponentiation and modulus multiplication. BGN cryptosystem [1] is employed in [2,14], which also requires a large amount of computation resource. Commutative Encryption Function [18] is used in [20], which also needs a lot time to do the encryption. In addition, Zhu et al. adopted the confusion matrix model, however, she only considered the friends that can be matched within 1-hop range, which may not be the best match.

To reduce the computation cost in existing protocols and not rely on TTP, according to nonhomomorphic encryption-based privacy-preserving scalar product computation [9], we propose a novel multi-hop protocol (NMHP) by employing confusion matrix transformation algorithm instead of complex computations. The main contributions of this paper are shown as follows.

(1) We propose TTP-free protocol (NMHP), in our protocol, we protect users' profile item details during the matching.
(2) We use the lightweight confusion matrix transformation algorithm instead of public-key cryptosystem and homomorphic encryption in NMHP.
(3) We consider profile matching in multi-hop instead of one hop, in this way, we can find the better match within a wider range.

The rest of this paper is organized as follows. Section 2 shows some preliminaries in our work. Following in Sect. 3, we describe the details of our proposed protocol and prove the correctness of our protocol. Sections 4 and 5 give the security analysis and performance evaluation. Finally, we draw the conclusions in Sect. 6.

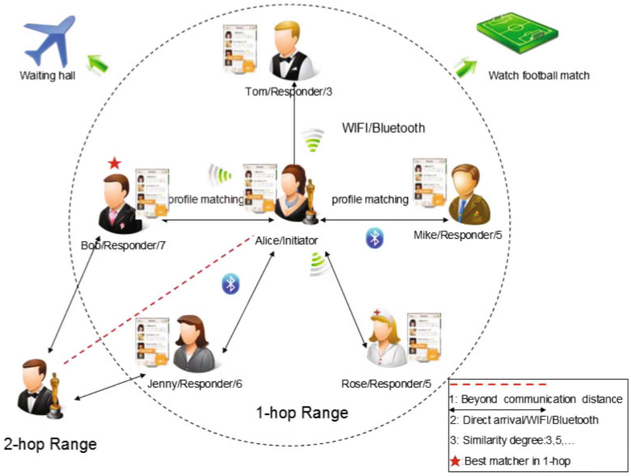

Fig. 2. Multi-hop profile matching process

2 Relate Definition

2.1 System Model

The whole process can be divided into two phases, illustrated by Figs. 1 and 2.

(1) The 1-Hop Friend Discovery Matching Phase. When user joins social networks, they usually begin by creating a profile, and then interact with other users. He/She will broadcast his/her personality profile in the list of neighbors within the 1-hop range. In the process, we mainly use the dot product and matrix confusion algorithm to obtain the degree of similarity between two users. In this way, we can find the *initiator*'s best matcher user in the 1-hop range.

(2) The Multi-hop Friend Discovery Matching Phase. Actually, there is often no suitable friend in 1-hop, so we need to use the multi-hop ideas in our protocol. We use *responder* as an agent to forward the *initiator*'s personality profile and calculate the degree of similarity between the next hop or multi-hop user.

2.2 Adversary Model

In the profile matching phase, if a party obtains one or more users (partial or full) attribute sets without their explicit consents, it is said to conduct user profiling attacks [9]. In this paper, we consider two types of user profiling attacks.

(1) Honest but Curious Adversaries Model. The honest but-curious (HBC) adversary is a legitimate participant in a communication protocol who will not deviate from the defined protocol but will attempt to learn all possible information from legitimately received messages. In this paper, we assume that the attacker is more interested in the privacy of mobile social network actors.

(2) Malicious Adversaries Model. A user who may launch some active attacks do not honestly follows the protocol but tries to learn more information than allowed. These adversaries may behave arbitrarily and are not bound in any way to follow a specified protocol. Such as denial-of-service (DoS) and continuous fake-profile attacks.

2.3 Design Goal

Our main goal is to achieve a secure private matching between an *initiator* and several *responder*s.

(1) Definition 1. Level-I Privacy. When the protocol process ends, the *initiator* and each *responder* should only know the size of the intersection set of their attributes (the attribute degree of similarity) set mutually. Any other attribute information should not be known by any other party.

(2) Definition 2. Level-II Privacy. When the protocol process ends, it is assumed that the outside attacker can intercept the user's interaction information, but the outside attacker can not decrypt or recover the user's details.

3 Friend Discovery Privacy Preserving Profile Matching

In this section, we propose our profile matching protocol in privacy-preserving proximity-based mobile social networks. As shown in Fig. 2. Firstly, we will introduce the basic idea behind the proposed protocol. Then, we will introduce the

Table 1. Summary of notations

Notation	Description
$MA_{l \times n}$, $MA^{*}_{l \times n}$	The *initiator* profile matrix and confusion profile matrix
$MC_{l \times n}$, $MD_{l \times n}$	Randomly generate two matrixes
α, β	Two large prime
$E()$	Asymmetric encryption function
PK, SK	Public key and private key
$H()$	Hash function
$\overrightarrow{K}$	The secret key to help get original results
$MB_{l \times n}$, $MB^{T}_{l \times n}$	The *responder* profile matrix, the transpose of $MB_{l \times n}$
D	The product of matrix MA^{*} and MB^{T}
$(W_{ij})_{l \times l}$	The weight degree between *initiator* and *responder*
λ_{max}	The similarity between the *initiator* and *responder*
$\lambda^{/}_{max}$	The similarity between the *initiator* and *responder* in n-hop
MSG_{I2R}	MSG from the *initiator* to the *responder*
MSG_{R2I}	MSG from the *responder* to the *initiator*
MSG_{A2R}	MSG from the *agent* to the *responder*

protocol in details. It mainly consists of the following three phases: system initialization, the 1-hop friend discovery matching phase, and the multi-hop friend discovery matching phase. The summary of notations used in our protocol is shown in Table 1.

3.1 The System Initialization Phase

During this stage *Alice* and other users who are willing to participate in social activities will start installing applications on their Smartphone or Tablet, in our designed mechanism, this application includes a number of functional modules each functional module will correspond to an actual application scenario and a user list. In a scenario, social users can choose their own interests and preferences for variety of presetting options, thus forming a property of profile, then *Alice* broadcast the profile to the her own user list, to start with interesting social activities. Specific friend discovery profile matching process see Sect. 3.2.

3.2 The 1-hop Friend Discovery Matching Phase

Alice first runs the following steps to start the preparation of the matching (see Fig. 3).

Step 1-Matrix-Confusion. In our protocol, we assume the PMSNs application developers define a public attribute set in advance, which consists of n attributes

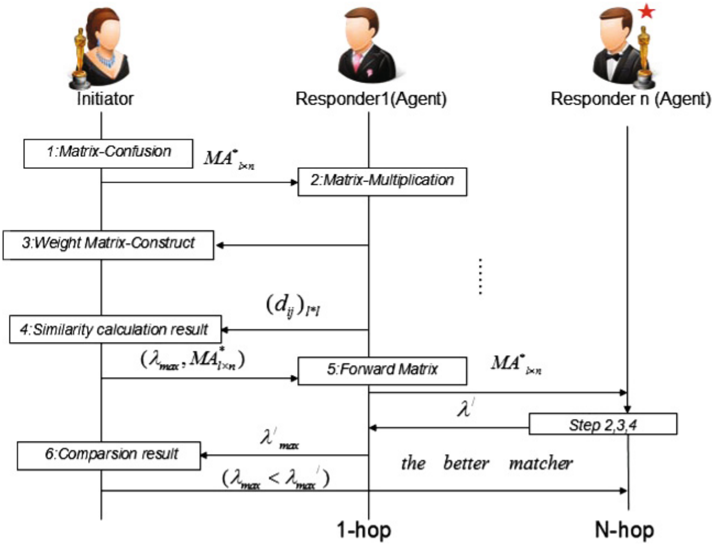

Fig. 3. Working processes of NMHP

$I = \{I_1, I_2, ...I_n\}$, when a user first joins the application, he/she will select a corresponding integer $i \in [1, l]$, (l called as the weight of an attribute) for each attribute to indicates the level of interest. l could be a small integer, say 2–10, which may be sufficient to differentiate user's interest level. Given n user attribute and l kinds of user interest level weight, we can organize the set of rating scores as a $n - by - l$ rating matrix $MA_{l\times n}$, this matrix can completely describe an user's profile, in which the row vectors indicate the weight of interest and column vectors mean the public attribute.

Alice first chooses a subset of $I_{Alice} = \{I_{ai1}, I_{ai2}, ...I_{aij}\}$ to indicate the profile items she wants to match with *Bob*, i.e., $MA_{l\times n}$, where a_{ij} denotes the user's interest level of the $j'th$ attribute in the public attribute set is i, $a_{ij} \in [0, 1]$. For example, if the *Alice*'s interest level of the $5'th$ attribute in the public attribute set is 3, he set $a_{35} = 1$ and $a_{n5} = 0$.

$$MA_{l\times n} = \begin{bmatrix} a_{11} & a_{12} & ... & a_{1n} \\ a_{21} & a_{22} & ... & a_{2n} \\ ... & ... & ... & ... \\ a_{l1} & a_{l2} & ... & a_{ln} \end{bmatrix}$$

Secondly, she choose two large prime α, β, where $|\alpha| = 256, \beta > (n+1)l^2\alpha^2$. Meanwhile, *Alice* randomly generate two matrixes $MC_{l\times n}$ and $MD_{l\times n}$, used for hiding personal information, computes the two large prime with $MA_{l\times n}$ and gets $MA^{*}_{l\times n}$. After that, *Alice* sends Msg_{I2R} as a matching query to *Bob* through her mobile device. The messages from *initiator* to the *responder* are:

$$Msg_{I2R} = \{MA^{*}_{l\times n}, ID_i, H(D_i), Q, t\} \quad (1)$$

where $MA^*_{l\times n}$ is a confusion matrix, which contains the weight of interest and public attribute of *initiator*. ID_i is the identity of the *initiator* (e.g., IP address), $H(ID_i)$ stands for the hash value of ID_i. Q denotes friend discovery query, and t refers to the time point at which the user gets the result from the *responder*. If beyond t, this information will be abandoned or relaunched.

Step 2-Matrix-Multiplication. After receiving *Alice*'s matching query $Msg_{_{I2R}}$, *Bob* dose the following: Firstly, *Bob* will use the public hash function $H()$ to check the correctness of *Alice*'s identity information. i.e., $H()+ID_i = H(ID_i)$. Secondly, if *Bob* (or other *responder*s) is also interested in profile matching, then he/she will choose a subset of $I_{Bob} = \{I_{bi1}, I_{bi2}, ...I_{bin}\}$ to indicate the profile items and construct a *responder* matrix that he wants to match with *Alice*, i.e., $MB_{l\times n}$, and computes the intersection of $MA_{l\times n}{}^*$ and $MB^T_{l\times n}$. If they have same interest, the value will be 1, or will be 0. When the process ends, *Bob* gets a matrix $D = MA^*{}_{l\times n} * MB^T_{l\times n} = (d_{ij})_{l*l}$. Lastly, *Bob* sends $Msg_{_{R2I}}$ as a reply of the matching to *Alice* through his mobile device. The messages from the *responder* to *initiator* are:

$$Msg_{_{R2I}} = \{D, H(D), ID_R, H(ID_r), PK_{Bob,}t\} \tag{2}$$

where ID_r is the identity of the *responder*, The public key is the groundwork for the following task. $H(ID_r)$, $H(D)$ are ID_r's, D's hash value, mainly used to verify the value of ID_r and D whether being modified by the external attackers in the transmission process.

Step 3-Weight Matrix-Construct. When *Alice* receives *Bob* message $Msg_{_{R2I}}$, she will verity the message. After verifying, *Alice* will construct a weight matrix $(W_{ij})_{l\times l}$, the weight matrix can indicate the different attention (interest) degree for the attributes between *initiator* and *responder*, we describe the generation of the element $(W_{ij})_{l\times l}$ according to the Formula-3:

$$(W_{ij})_{l\times l} = \begin{cases} i; & i = j; \\ i - |i-j|; & (i - |i-j|) > 1; \\ 1; & (i - |i-j|) \leq 1; \end{cases} \tag{3}$$

Step 4-Similarity Calculation Results. In this step, *Alice* runs Algorithm 1 to get T^*. Meanwhile, *Alice* uses T^* and $(W_{ij})_{l\times l}$ to make dot production. Up to now, *Alice* can get the match value λ_{Bob}, then she knows the similarity with *Bob*. And so on, *Alice* can knows the similarity value with the other *responder* in 1-hop. i.e., $\lambda = \begin{bmatrix}\lambda_1 \ \lambda_2 \ ... \ \lambda_n\end{bmatrix}$. Until now, *Alice* can choose the user who has the largest similarity value λ_{max} in the 1-hop region as the best matcher.

We assume *Bob* is the best match user, he also wants to make friends with *Alice*, so *Alice* will start a session process and send a message to *Bob*.

$$Msg_{_{I2R}} = \{PK_{Alice}, E_{PK_{Bob}}(C)\} \tag{4}$$

where $E_{PK_{Bob}}(C)$ is cipher text which is encrypted by *Bob*'s public key, C is a specific session content (plain text).

Algorithm 1. Similarity Calculation Results Algorithm

```
Input:
    $Alice$ received $Bob$ computer results $D = (d_{ij})_{l*l}$;
Output:
    Generate similarity value $\lambda$ by $Alice$ and $Bob$'s matrix dot production;
$Alice$ computer $T = (t_{ij})_{l*l}$, $t_{ij} = (d_{ij} + k_i) \bmod \alpha$, $for\, d_{ij} \in D_{l\times l}$;
$Alice$ makes a further transformation to get $T^* = \frac{t_{ij}-(t_{ij} \bmod \alpha^2)}{\alpha^2}$;
$Alice$ considers the corresponding weights and computes $H_{l\times l} = T^*_{l\times l} \circ (W_{ij})_{l\times l}$, "$\circ$" is dot production;
Up to now,we can get the match (similarity) value $\lambda = \sum_{i=1}^{l}\sum_{j=1}^{l} h_{ij}$;
return $\lambda$.
```

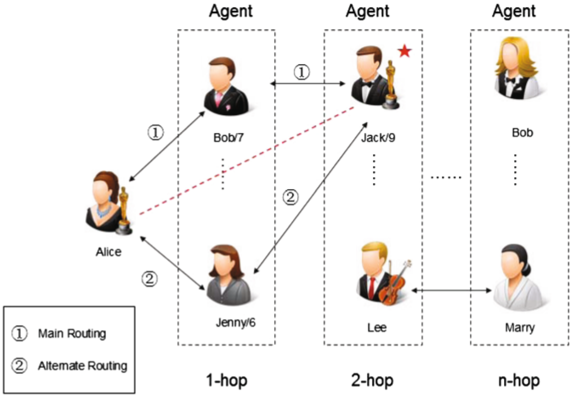

Fig. 4. Multi-hop friend discovery profile matching process

3.3 The Multi-hop Friend Discovery Matching Phase

Step 5-Forward the Initiator Profile Matrix. In the actual scenario, in 1-hop range the chances of finding a friend who has the similar interests and hobbies are usually limited, so we use the idea of multi-hop in the network, use a responder as a agent to forward the initiator profile to find the better match user, then we may find the largest similarity degree friend with *Alice* in a wider range (see Fig. 4).

In this section, in order to simply describe the process, we set the routing as two hops, *Alice* as a initiator, *Bob* and *Jenny* are *Alice*'s neighbors in 1-hop range, *Alice* can not directly through the Wi-Fi and Bluetooth communicate with *Jack* (beyond the range of communication), *Bob*, *Jenny*, and *Jack* can communicate directly (they are neighbors in a hop range); Firstly, in order to expand the range of friend discovery, *Alice* will randomly select a user (we assume the user is *Bob*) from her user list as a agent for forwarding her profile ($MA^*{}_{l\times n}$) and the max similarity value (λ_{max}); Secondly, *Jack* receives this profile, he will carry out matrix calculation; Lastly, *Bob* will calculate the similarity result with dot product. This process is the same as *Step 2, 3, 4*, the messages from the agent (*Bob*) to *responder* (*Jack*) are:

$$Msg_{A2R} = \{\lambda_{max}, MA^*_{l\times n}, ID_{agent}, t\}. \tag{5}$$

Step 6-Comparison of Results. Via calculation *Bob* can get the match value λ_{Jack}, then *Bob* knows the similarity with *Jack*, and he will compare λ_{Jack} with λ_{max}, when $\lambda_{jack} > \lambda_{max}$, *Bob* will forward the result to *Alice*. So far, *Alice* decides whether or not to build up a communication relationship with *Jack*.

Otherwise *Alice* knows that the search for a matching user has failed in the 2-round. And she will choose the best match user in the first round or continue to search for matching users in the next hop.

4 Security Analysis

4.1 Resistance to Attacks from HBC Model

(1) Privacy of the *Initiator*. In our protocol, *Alice* does not directly send her interested profile to *Bob*, The *initiator* reveals his/her personal profile matrix $MA^*_{l\times n}$ to the potential *responder*s in vicinity, all the elements in this transition matrix $MA^*_{l\times n}$ are randomly generated by two large prime α, β, meanwhile, we use the random matrix $MC_{_{l\times n}}$,$MD_{_{l\times n}}$ to make a confusion about the original matrix, and each of random number in the matrix is used only once in the interactive process. So it is very difficult for the *responder* to recover the *initiator* profile. This method is proved to be secure in [9]. Therefore, except for the profile items *Alice* and *Bob* are both interested in, the names of other unnecessary items are all protected.

(2) Privacy of the *Responder*. In our protocol, elements in matrix $MB_{l\times n}$ is also privacy-preserving using the large prime β and matrix multiplication, obviously, the unknown a^*_{ix} will hide the operation on each $D = (d_{ij})_{l*l}$, although *Alice* can decrypt the data *Bob* sends to him, say $Msg_{_{R2I}}$, because of the determined operation properties of the matrix, *Alice* will not monitor the matching process and decrypt the intermediate results to get the original results of $MA_{l\times n} * MB^T_{l\times n}$, so *Alice* learns nothing about the *responder* other than the matching value. The privacy of the *responder* can be protected too. Moreover, any other sensitive information (e.g., *responder* attributes, interest level etc.) cannot be inferred from the exponentiated values they receive.

4.2 Defense Against Outside Adversary

If external adversary would like to use the background knowledge to build attack dictionary to attack on the user, then this dictionary will be very large to make the dictionary profiling. This is due to the attribute could be the user's personal information including profession, interest, favorite song, etc. So in most cases, it is very difficult for the external adversary trying to get the user's attribute information to get through the brute force.

In our protocol, all the delivered important information will be processed by a hash function H() to get a hash value, this value is also included in the message, when users get the message, they can check if the information is modified or not by comparing the hash value. Furthermore, we assume that an attacker intercepts information through an insecure channel. However, because we use asymmetric encryption algorithm, the external attacker does not know the user's private key, so he has no way to recover the information. Hence, our protocol can resist external attackers.

5 Performance Evaluation

In this section, we evaluate the performance of the proposed protocol in terms of computation complexity and communication overhead in the mobile social networks. At the same time, we compare our protocol's privacy preserving degree.

Table 2. Comparison of related work

Protocol	Offline		Online		Communication.(bits)	
	Initiator	Responder	Initiator	Responder	Initiator	Responder
NMHP	$l\cdot\mathrm{n}\cdot\mathrm{mul1}+ 2l\cdot\mathrm{n}\cdot add$	~	$2l\cdot\mathrm{l}\cdot\mathrm{mul1}+ 3l\cdot\mathrm{l}\cdot\mathrm{n}\cdot add$	$2l\cdot\mathrm{l}\cdot mul1+ 3l\cdot\mathrm{l}\cdot\mathrm{n}\cdot add+ l.n.mul1+ l.n.add$	$(l\cdot+2)\cdot 1024$	$2(l\cdot\mathrm{l})\cdot 1024$
WAS [11]	$n\cdot exp1+ n\cdot h$	$n\cdot exp1+ n\cdot\mathrm{h}$	$n\cdot exp1$	$n\cdot exp1$	$2n\cdot 1024$	$(n+2)\cdot 1024$
Fine-grained [19]	$2l\cdot\mathrm{n}\cdot exp1+ l\cdot\mathrm{n}\cdot mul2$	~	$1\cdot exp2$	$1\cdot exp1+ 1\cdot exp2+ n\cdot mul2$	$l\cdot\mathrm{n}\cdot 2048$	$l\cdot 2048$

We make an analysis about the complexity of our protocol and some existing work in this section. The offline, online computation cost as well as the communication overhead are used to measure the complexity of our protocol. The number of the multiplication and exponentiation operations is used to evaluate the computation cost, since these operations are always resource-consuming in mobile devices. The communication overhead is evaluated by counting the transferred and received bits.

In our paper *exp1* means 1024-bit exponentiation operation, *exp2* means 2048-bit exponentiation operation, *add* indicates modular addition, and *mul1, mul2* represent 1024-bit and 2048-bit multiplication operation, respectively. We assume that each user in our protocol has n interests and the highest corresponding weight value is l. From Table 2, we can learn that our protocol decreases computation and communication costs significantly, especially the online computation cost which has a direct impact on system performance.

6 Conclusion

In this paper, we have proposed a multi-hop profile matching protocol for privacy preserving in mobile social networks. By computation of profile similarity, users can find out potential friends with similar interests, skills, age, location, etc., through a privacy preserving way. Our protocol uses confusion matrix technology, dot product, weight of interest level to get the similarity value, in which the weights and threshold are both chosen by users themselves. Detailed security analysis shows that the privacy of both names and values of users' profile items is well protected by our protocol.

Acknowledgments. This work is supported in part by the National Natural Science Foundation of China under Grant Numbers 61272151, 61472451 and 61402161, the International Science & Technology Cooperation Program of China under Grant Number 2013DFB10070, the China Hunan Provincial Science & Technology Program under Grant Number 2012GK4106, the Hunan Provincial Education Department of China under grant number 2015C0589. and the "Mobile Health" Ministry of Education - China Mobile Joint Laboratory (MOE-DST No. [2012]311).

References

1. Boneh, D., Goh, E.-J., Nissim, K.: Evaluating 2-DNF formulas on ciphertexts. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 325–341. Springer, Heidelberg (2005)
2. De Cristofaro, E., Kim, J., Tsudik, G.: Linear-complexity private set intersection protocols secure in malicious model. In: Abe, M. (ed.) ASIACRYPT 2010. LNCS, vol. 6477, pp. 213–231. Springer, Heidelberg (2010)
3. Dong, W., Dave, V., Qiu, L., Zhang, Y.: Secure friend discovery in mobile social networks. In: 2011 IEEE INFOCOM, pp. 1647–1655. IEEE (2011)
4. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient private matching and set intersection. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 1–19. Springer, Heidelberg (2004)
5. Ioannidis, I., Grama, A., Atallah, M.J.: A secure protocol for computing dotproducts in clustered and distributed environments. In: Proceedings of the 2002 International Conference on Parallel Processing, pp. 379–384. IEEE (2002)
6. Kissner, L., Song, D.: Privacy-preserving set operations. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 241–257. Springer, Heidelberg (2005)
7. Li, M., Cao, N., Yu, S., Lou, W.: Findu: privacy-preserving personal profile matching in mobile social networks. In: 2011 IEEE INFOCOM, pp. 2435–2443. IEEE (2011)
8. Lu, R., Lin, X., Liang, X., Shen, X.: A secure handshake scheme with symptoms-matching for mhealthcare social network. Mob. Netw. Appl. **16**(6), 683–694 (2011)
9. Lu, R., Lin, X., Shen, X.: Spoc: a secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency. IEEE Trans. Parallel Distrib. Syst. **24**(3), 614–624 (2013)
10. Manweiler, J., Scudellari, R., Cox, L.P.: Smile: encounter-based trust for mobile social services. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, pp. 246–255. ACM (2009)

11. Niu, B., Zhu, X., Liu, J., Li, Z., Li, H.: Weight-aware private matching scheme for proximity-based mobile social networks. In: 2013 IEEE Global Communications Conference (GLOBECOM), pp. 3170–3175. IEEE (2013)
12. Pietiläinen, A.K., Oliver, E., LeBrun, J., Varghese, G., Diot, C.: Mobiclique: middleware for mobile social networking. In: Proceedings of the 2nd ACM Workshop on Online Social Networks, pp. 49–54. ACM (2009)
13. Rane, S., Sun, W., Vetro, A.: Privacy-preserving approximation of l1 distance for multimedia applications. In: 2010 IEEE International Conference on Multimedia and Expo (ICME), pp. 492–497. IEEE (2010)
14. Sang, Y., Shen, H.: Efficient and secure protocols for privacy-preserving set operations. ACM Trans. Inf. Syst. Secur. (TISSEC) **13**(1), 9 (2009)
15. Von Arb, M., Bader, M., Kuhn, M., Wattenhofer, R.: Veneta: serverless friend-of-friend detection in mobile social networking. In: 2008 IEEE International Conference on Wireless Communications, Networking and Mobile Computing, pp. 184–189. IEEE (2008)
16. Wang, Y., Vasilakos, A.V., Jin, Q., Ma, J.: Survey on mobile social networking in proximity (msnp): approaches, challenges and architecture. Wireless Netw. **20**(6), 1295–1311 (2014)
17. Wang, Y., Xu, J.: 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), pp. 133–140. IEEE (2014)
18. Xie, Q., Hengartner, U.: Privacy-preserving matchmaking for mobile social networking secure against malicious users. In: 2011 Ninth Annual International Conference on Privacy, Security and Trust (PST), pp. 252–259. IEEE (2011)
19. Zhang, R., Zhang, J., Zhang, Y., Sun, J., Yan, G.: Privacy-preserving profile matching for proximity-based mobile social networking. IEEE J. Sel. Areas Commun. **31**(9), 656–668 (2013)
20. Zhu, X., Liu, J., Jiang, S., Chen, Z., Li, H.: Efficient weight-based private matching for proximity-based mobile social networks. In: 2014 IEEE International Conference on Communications (ICC), pp. 4114–4119. IEEE (2014)