IEEE TRANSACTIONS ON
EMERGING TOPICS
IN COMPUTING

Received 4 April 2013; revised 11 July 2013; accepted 15 July 2013. Date of publication 26 August 2013;
date of current version 20 September 2013.

Digital Object Identifier 10.1109/TETC.2013.2279541

Fairness-Aware and Privacy-Preserving Friend Matching Protocol in Mobile Social Networks

HAOJIN ZHU[1] (Member, IEEE), SUGUO DU[2], MUYUAN LI[1] (Student Member, IEEE), AND ZHAOYU GAO[1] (Student Member, IEEE)

[1]Computer Science Department, Shanghai Jiao Tong University, Shanghai 200030, China
[2]Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai 200030, China
CORRESPONDING AUTHOR: S. DU (sgdu@sjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grants 61003218, 70971086, 61272444, 61161140320, and 61033014, and by the Doctoral Fund of Ministry of Education of China under Grant 20100073120065.

ABSTRACT Mobile social networks represent a promising cyber-physical system, which connects mobile nodes within a local physical proximity using mobile smart phones as well as wireless communication. In mobile social networks, the mobile users may, however, face the risk of leaking their personal information and location privacy. In this paper, we first model the secure friend discovery process as a generalized privacy-preserving interest and profile matching problem. We identify a new security threat arising from existing secure friend discovery protocols, coined as runaway attack, which can introduce a serious unfairness issue. To thwart this new threat, we introduce a novel blind vector transformation technique, which could hide the correlation between the original vector and transformed results. Based on this, we propose our privacy-preserving and fairness-aware interest and profile matching protocol, which allows one party to match its interest with the profile of another, without revealing its real interest and profile and vice versa. The detailed security analysis as well as real-world implementations demonstrate the effectiveness and efficiency of the proposed protocol.

INDEX TERMS Privacy preserving, friend discovery, mobile social networks.

I. INTRODUCTION

Mobile online social networks have gained tremendous momentum in the recent years due to both the wide proliferation of mobile devices such as smartphones and tablets as well as the ubiquitous availability of network services. Moreover, the positioning technologies such as GPS, and Wireless localization techniques for mobile devices have made both the generation and sharing of real-time user location updates readily available. This, in turn, leads to the extreme popularity of location-aware social networks such as Foursquare [1], Gowalla [2] and Wechat, which boast up to hundreds of millions of users. Location-aware mobile social networks represent a promising Cyber-Physical System (CPS), which connects mobile nodes within a local physical proximity by using mobile smart phones as well as wireless communication. As a valuable complement to web-based online social networking, location-aware mobile social networks allow mobile users to have more tangible face-to-face social interactions in public places such as bars, airports, trains, and stadiums [3]. Profile matching is more than important for fostering the wide use of mobile social networks because finding the nearby individuals of the similar interests is always the first step for any social networking.

The existing mobile social network systems pay little heed to the security and privacy concerns associated with revealing one's personal social networking preferences and friendship information to the ubiquitous computing environment. In particular, in mobile social networks, the mobile users may face the risk of leaking of their personal information and their location privacy. Under this circumstance, the attackers can directly associate the personal profiles with real persons nearby and then launch more advanced attacks. Existing researches show that loss of privacy can expose users to unwanted advertisement and spams/scams, cause social

reputation or economic damage, and make them victims of blackmail or even physical violence [4].

Recently, there are quite a few proposals for *Private Profile Matching*, which allow two users to compare their personal profiles without revealing private information to each other [5], [6]. In a typical private profile matching scheme, the personal profile of a user consists of multiple attributes chosen from a public set of attributes (e.g., various interests [5], disease symptoms [7], or friends [8] etc.). The private profile matching problem could then be converted into Private Set Intersection (PSI) [9], [10] or Private Set Intersection Cardinality (PSI-CA) [11], [12]. In particular, two mobile users, each of whom holds a private data set respectively, could jointly compute the intersection or the intersection cardinality of the two sets without leaking any additional information to either side.

However, there are quite a few challenges which make the existing private profile matching solutions less practical in applications. For example, similar to most of the online social network applications, a mobile social networking user is expected to freely search its potential common-interest friends by matching his *interest* with the *personal profiles* of the searching targets rather than making the profile matching directly. As is shown in Fig. 1, Alice has her personal profile,

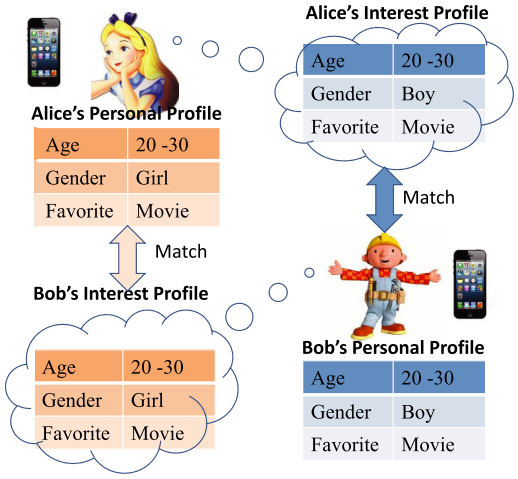


FIGURE 1. Friend discovery in mobile social networks.

which includes three attributes: age, girl and movie. She is interested in finding a boy with similar age and hobbies. Conversely, Bob also has his own profile and interests. A successful matching could be achieved in case that Alice's profile matches Bob's interest while, at the same time, Bob's profile matches Alice's interest. Such a mapping process could be well supported by the existing online dating social networks, in which a member may seek another member satisfying some particular requirements (e.g., gender, age ranges or even living location as in [13]). Further, the existing proposals are one-way only and profile matching requires running a protocol twice, with reversed roles in the second run. This two-pass protocol may be exploited by the dishonest user or even a malicious attacker to launch the *runaway attack*, in which a malicious one that wants to learn another user's interests but is unwilling to reveal his own interests can simply abort the protocol in the second round. This runaway attack incurs a serious unfairness issue. The runaway attack may be more challenging in the case of separating user's profile from his interest since matching the users' profile and the interest could only be achieved in two steps.

To solve the above mentioned challenges and thus further enhance the usability of mobile social networks, we present a novel Privacy Preserving and Fairness-aware Friend Matching Protocol. In the designed protocol, a successful matching only happens in case that the interests of both of the participants could match the profiles of the others. In other words, no one can learn any extra information from the protocol unless another participant is exactly what he is looking for and vice versa. Our work is motivated from a simple observation that if two vectors match, they will still match no matter whether they are transformed in the same way (e.g., add or remove a randomly generated vector) or shuffled with the same order.

To achieve this goal, we introduce a novel *Blind Vector Transformation* technique, under which each participant contributes a part of the vector transformation while any single one of the parties cannot recover the original vectors from the final transformation result. Therefore, with blind vector transformation, we could enable a party to match its interests with another's profile but, at the same time, to keep the interests as well as the profiles private. Further, to thwart runaway attack, we introduce a lightweight verifier checking technique, which enables the verifier to check the matching at the minimized overhead and prevent any participant from launching the runaway attack.

The contribution of this work could be summarized as follows:

- For the first time, we separate the user's interest from its profile, which is expected to be a generalization of traditional profile matching problem.
- We introduce a novel blind vector transformation technique, which could hide the correlation between the original vector and the transformed result. Based on it, we propose the privacy-preserving and fairness-aware friend matching protocol, which enables one party to match its interest with the profile of another, and vice versa, without revealing their real interest.
- We introduce a novel lightweight verifier checking approach to thwart runaway attack and thus achieve the fairness of two participants.
- We implement our protocols in real experiments. We demonstrate the performance of the proposed scheme via extensive experiment results.

The remaining of this paper is organized as follows. In Section II, we introduce the system model, adversary model as well as the designing objectives. In Section III, we presents the privacy-preserving and fairness-aware friend matching protocol. In Section IV, we give a detailed

analysis on the security of the proposed protocol. In Section V, the performance of the proposed protocol is validated by extensive experiments and analysis, which is followed by the conclusion and future work in Section VII.

II. SYSTEM, ADVERSARY MODEL AND PRELIMINARIES

In this section, we first introduce our system model as well as the adversary model. Then we present our design goals. Before introducing the proposed protocol, we also give a brief introduction on some cryptographic foundations, including Paillier Homomorphic Encryption.

A. SYSTEM MODEL

In mobile social networks, a user launches a query to find the potential friends, when he comes to a new places. Before the query, a user should initialize a profile as his inherent characteristic. This profile consists of multiple attributes (e.g., user's occupation, hobbies and other private information), which could be denoted as a vector $\mathbb{P} = \{p_1, p_2, \ldots, p_n\}$. Here, $p_j (j = 1, \ldots, n)$ is an integer, which refers to an attribute of $\mathbb{P}$. When a user issues a query, he firstly generates the corresponding interest vector $\mathbb{I} = \{i_1, i_2, \ldots, i_n\}$. Note that, similar to a typical search process of online social networks, the user could freely generate different interests for multiple times.

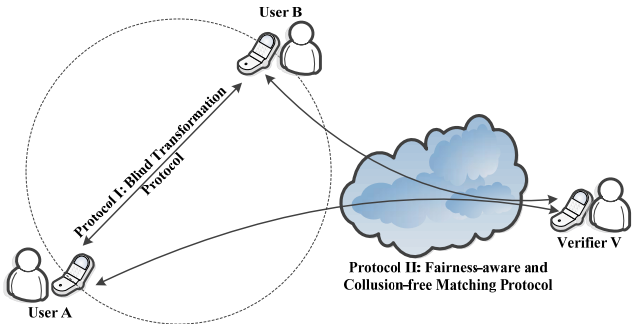


FIGURE 2. **System architecture.**

A typical friend discovery process could be described as follows. User A will send his current interest $\mathbb{I}_A$ to user B, and then he will obtain B's current interest $\mathbb{I}_B$. After the interests are exchanged, A will compare his own profile $\mathbb{P}_A$ with $\mathbb{I}_B$ while B compares his profile $\mathbb{P}_B$ with $\mathbb{I}_A$. We define a successful matching as $\mathbb{P}_A$ matches $\mathbb{I}_B$ and, at the same time, $\mathbb{P}_B$ matches $\mathbb{I}_A$, which is similar to the privacy level introduced in [5].

Note that, in some cases, the user only cares about some specific attributes. To deal with those fields that are not considered in one query, we separate the fields of $\mathbb{I}$ into interested fields (IF) and non-interested fields (NIF). Therefore, a successful matching should ensure the IF fields of the interests and profiles match while whether NIF fields match or not cannot affect the final comparison results. To achieve this, to make NIF fields not affect the comparison results, a very large value is assigned to the NIF fields of the interest, which will directly make the comparison fail no matter what the value of the profiles is.

We also assume the existence of a randomly chosen verifier. This verifier is either honest, semi-honest or actively malicious. We also assume that the verifier could launch the collusion attack by collaborate with one of two friend discovery users.

B. ADVERSARY MODEL

The adversary is considered to be curious with others' profile and interest. Therefore, if without an appropriate security countermeasure, the friend discovery process may suffer from a series of privacy threats. In particular, we consider the following adversary model:

1) **Privacy Inference from Profile Matching:** The adversary tries to find out the interests or the profiles of the other users during the profile matching process.
2) **Privacy Inference from Aborting the Protocol (Runaway Attack):** Under this attack, even with a privacy-preserving profile matching protocol, the adversary aims to infer the private information of another user by stopping the protocol during the friend matching process and performing certain analysis over the information already obtained. This attack will introduce a serious unfairness issue since, in a two-pass protocol, the adversary could refuse to send his matching result after obtaining the result from his partner.
3) **Collusion Attack:** The adversary may collude with other users to infer the user's private information.

C. DESIGNING OBJECTIVES

The proposed Privacy-preserving and Fairness-aware Friend Matching Protocol should satisfy the following objectives:

1) **Privacy Guarantee:** The protocol should support profile matching of mobile users without leaking mobile users' private information. In particular, no attackers including the external and internal attackers could obtain the profile or interest information of the users. In the proposed protocol, after performing the privacy-preserving friend matching protocol, each participant could only obtain the comparison result "success" or "fail". No other information will be leaked from the protocol.
2) **Fairness Assurance:** In each phase of the protocol, a user can obtain personal information from others as much as his own personal information leaking from the protocol. In other words, no one can gain more information than what he tell others, which ensures the fairness of the protocol.

D. PAILLIER HOMOMORPHIC ENCRYPTION

The protocol proposed in this paper is based on Paillier's homomorphic encryption. In the follows, we summarize how

Paillier cryptosystem works to help illustrate and understand our protocols.

- **Key Generation**: The trusted third party chooses two large prime numbers p and q randomly such that $gcd(pq, (p-1)(q-1)) = 1$ and compute $n = pq$ and $\lambda = \text{lcm}(p-1, q-1)$. It then selects a random $g \in \mathbb{Z}^*_{N^2}$ such that $\gcd(\text{L}(g^\lambda \bmod N^2), N) = 1$, where $\text{L}(x) = (x-1)/N$. The entity's Paillier public and private keys are $< N, g >$ and λ, respectively.
- **Encryption**: Let m be a message to be encrypted where $m \in \mathbb{Z}_n$ and $r \in \mathbb{Z}_n$ be a random number. The ciphertext could be given by

$$E(m \bmod N, r \bmod N) = g^m r^N \bmod N^2$$

  where $E()$ denotes the Paillier encryption operation.
- **Decryption**: Given a ciphertext $c \in \mathbb{Z}^*_{N^2}$, the corresponding plaintext can be derived as

$$D(c) = \frac{\text{L}(c^\lambda \bmod N^2)}{\text{L}(g^\lambda \bmod N^2)} \bmod N$$

  where $D()$ denotes the Paillier decryption operation using private key $sk = \lambda$ hereafter.
- **Homomorphic**: Given $m_1, m_2, r_1, r_2 \in \mathbb{Z}_N$, it satisfies the following homomorphic property:

$$E(m_1) \cdot E(m_2) = E(m_1 + m_2)$$

In this study, for simplicity of presentation, we denote the encryption of profile $\mathbb{P} = (p_1, p_2, \ldots, p_n)$ under public key k as $\texttt{Encrypt}(\mathbb{P}, k) = (\texttt{Encrypt}(p_1, k), \texttt{Encrypt}(p_2, k), \ldots, \texttt{Encrypt}(p_n, k))$ while $\texttt{Encrypt}(\mathbb{P}, k)$ is the decryption function, the same with interest $\mathbb{I}$. We assume that each user i has his own Paillier public and private key (pk_i, sk_i), and the encryption and decryption in this paper are all executed in Paillier cryptosystem.

III. THE PROPOSED PRIVACY-PRESERVING AND FAIRNESS-AWARE FRIEND MATCHING PROTOCOL

In this section, we will present the details of our protocols. Firstly, we will introduce the basic idea behind the proposed protocol. We will then introduce the protocol in details.

A. PROTOCOL OVERVIEW

As shown in Fig. 2, the proposed protocol is comprised of two basic protocols, including: Protocol I: Blind Vector Transforming Protocols; Protocol II: Fairness-aware and Collusion-free Matching Protocol. The basic idea of blind vector transformation protocol is allowing two untrusted parties to transform two vectors into the blind ones by following a series of private and identical steps, e.g., adding a random vector, shuffling in the same order. Since the transformation follows the same step, the matching results (e.g. the number of matched interest and profiles) keep unchanged before and after the transformation, which enable the untrusted participants compare the profile without leaking their real interest or profile information.

The major challenge of the blind vector is how to hide the real value of the interest or profile of the participants. The basic idea is that two untrusted participants will contribute a part of this transformation while each of them cannot recover the real interest or profile. To achieve this, we define five primitive operations as follows:

- **Encrypt**: Given a vector $\vec{v}$, it performs Paillier encryption on it with public key pk to obtain the ciphertext $E_{pk}[\vec{v}]$. Such an operation is denoted as $\texttt{Encrypt}(\vec{v}, pk)$.
- **Vecadd:** Given two vectors $\vec{r}$ and $\vec{v}$, both of which are encrypted under Paillier encryption, the operation VecAdd will be executed to perform a sum operation $E[\vec{v}]E[\vec{r}] = E[\vec{v} + \vec{r}]$. Such an operation is denoted as $\texttt{VecAdd}(\vec{v}, \vec{r})$.
- **Vecext:** Given a vector $\vec{v}$, the operation $\texttt{VecExt}(\vec{v}, \vec{r})$ could hide the real value of $\vec{v}$ by performing a diffusion operation, which appends some dummy vectors $\vec{r}$ to $\vec{v}$ to obtain $v||r$.
- **Vecshuffle**: Given a vector $\vec{v}$, the operation $\texttt{VecShuffle}(\vec{v})$ could hide the real value of $\vec{v}$ by performing a confusion operation, which randomly shuffles the elements in vector $\vec{v}$.
- **Vecrev**: Given a vector $\vec{v}$, the operation $\texttt{VecRev}(\vec{v}, k)$ could further hide the real value of $\vec{v}$ by randomly changing the value of k elements in vector $\vec{v}$.

We summarize 5 primitive operations in Table 1.

TABLE 1. Five primitive operations in blind transformation.

execution	function	operation
Encrypt	$\texttt{Encrypt}(\vec{v}, pk)$	Encrypt the vector $\vec{v}$ by Paillier Encryption with public key pk as $E_{pk}[\vec{v}]$
VecAdd	$\texttt{VecAdd}(\vec{v}, \vec{r})$	Add a random number vector $\vec{r}$ to $\vec{v}$. If both vectors are encrypted under Paillier Encryption, the adddition is performed as $E[\vec{v}]E[\vec{r}] = E[\vec{v} + \vec{r}]$
VecExt	$\texttt{VecExt}(\vec{v}, \vec{r})$	Append some dummy vector $\vec{r}$ to $\vec{v}$ to obtain $v\|\|r$.
VecShuffle	$\texttt{VecShuffle}(\vec{v})$	Randomly shuffle the elements in vector $\vec{v}$
VecRev	$\texttt{VecRev}(\vec{v}, k)$	Randomly change the value of k elements in vector $\vec{v}$

The overall procedure could be described as follows. In blind profile vector transformation phase, the user A encrypts his profile with his own public key by triggering operation $\texttt{Encrypt}(\vec{v}, pk)$. Here, Paillier is adopted since it keeps A's profile private and, at the same time, allows B to perform blind transformation on it. The transformation operations include VecAdd, VecExt, VecShuffle, VecRev. The user B also makes the same blind transformation on B's profile. After finishing these steps, in the matching phase, it is required that each participant should compare the blinded interest and profile. Each participant will send the number of matching vector pairs as well as the size of search interest to a verifier. The verifier will compare if the number of search interest equals to the number of matching vector pairs

Algorithm 1 The Blind Transformation Algorithm

Input: $\mathbb{P}'_a \leftarrow U_a$'s profile encrypted under his public key pk_a, $\mathbb{I}_b \leftarrow U_b$'s interest, $e_b \leftarrow$ the number of interest U_b considers in $\mathbb{I}_b$ and $l_b \leftarrow$ a security parameter.

Output: $\mathbb{P''}_a \leftarrow$ the blind-transformed profile vector for U_a, $\mathbb{I''}_b \leftarrow$ the transformed interest vector for U_b and $s_b \leftarrow$ the actual matching result for U_b.

function BLIND-TRANSFORMATION($\mathbb{P}'_a$, pk_a, $\mathbb{I}_b$, e_b, l_b)
- $r_b \leftarrow$ random vector of length $n = ||\mathbb{P}'_a||$
- $\widetilde{r'_b} \leftarrow \texttt{Encrypt}(r_b, pk_a)$
- $\widetilde{\mathbb{P}_a} \leftarrow \texttt{VecAdd}(\mathbb{P}'_a, r'_b)$
- $\widetilde{\mathbb{I}_b} \leftarrow \texttt{VecAdd}(\mathbb{I}_b, r_b)$
- $y_b \leftarrow$ random vector of length l_b
- $\widetilde{y'_b} \leftarrow \texttt{Encrypt}(y_b, pk_a)$
- $\widetilde{\mathbb{P}'_a} \leftarrow \texttt{VecExt}(\widetilde{\mathbb{P}_a}, y'_b)$
- $k_b \leftarrow$ random number between $[1, l_b]$
- $\widetilde{y_b} \leftarrow \texttt{VecRev}(y_b, k_b)$
- $\mathbb{I}'_b \leftarrow \texttt{VecExt}(\widetilde{\mathbb{I}_b}, \widetilde{y_b})$
- $\mathbb{I''}_b \leftarrow \texttt{VecShuffle}(\mathbb{I}'_b)$
- $\mathbb{P''}_a \leftarrow \texttt{VecShuffle}(\widetilde{\mathbb{P}'_a})$
- $s_b \leftarrow e_b + l_b - k_b$
- **return** $\mathbb{P''}_a, \mathbb{I''}_b, s_b$

end function

from both of parties. If both of them match, the verifier will inform *A* and *B* of a successful match. In this process, any participant will learn nothing about the personal information of another except match or not, which makes the proposed protocol achieve both of privacy preserving and fairness. In the follows, we will present the detailed protocol as follows.

B. SYSTEM INITIALIZATION PHASE

Without loss of the generality, we consider two nodes U_a and U_b for potential friend discovery. In the system initialization phase, the trusted third party will generate their private and public key pairs, which are denoted as (sk_a, pk_a) and (sk_b, pk_b), respectively. Their profiles are denoted as $\mathbb{P}_a$ and $\mathbb{P}_b$. For a matching, U_a and U_b may only consider e_a and e_b out of total n interest fields. Thus, there are $n - e_a$ and $n - e_b$ attributes which are excluded from this match. We assume the current interest vectors are $\mathbb{I}_a$ and $\mathbb{I}_b$.

C. THE PROPOSED BLIND TRANSFORMATION PROTOCOL

In the blind transformation phase, each participant will encrypt his profile by using his public key and provide it to his partner for blind transformation. In the follows, we introduce the blind transformation process by taking U_b transforming U_a's profile and his own interest as an example. It is similar for U_a to blind transform U_b's profile. U_a performs $\texttt{Encrypt}(\mathbb{P}_a, pk_a)$ to encrypt his profile $\mathbb{P}_a$, which is denoted as $\mathbb{P}'_a$. U_a sends $\mathbb{P}'_a$ and pk_a to U_b. Then, U_b performs the following blind transformation operations:

- **Blind Add:** U_b generates a random vector r_b, and then performs $r'_b = \texttt{Encrypt}(r_b, pk_a)$. After that, U_b calculates $\widetilde{\mathbb{P}_a} = \texttt{VecAdd}(\mathbb{P}'_a, r'_b)$ and $\widetilde{\mathbb{I}_b} = \texttt{VecAdd}(\mathbb{I}_b, r_b)$ by adding r'_b and r_b to $\mathbb{P}'_a$ and $\mathbb{I}_b$, respectively.
- **Blind Append:** U_b generates a random vector y_b of length l_b, where l_b is a predetermined security parameter, then performs $y'_b = \texttt{Encrypt}(y_b, pk_a)$ to get $\widetilde{\mathbb{P}'_a} = \texttt{VecExt}(\widetilde{\mathbb{P}_a}, y'_b)$.
- **Blind Reverse:** U_b randomly selects $k_b \in \{1, 2, \ldots, l_2\}$ and performs $\widetilde{y_b} = \texttt{VecRev}(y_b, k_b)$, then obtains $\mathbb{I}'_b = \texttt{VecExt}(\widetilde{\mathbb{I}_b}, \widetilde{y_b})$.
- **Blind Shuffle:** U_b performs $\mathbb{I''}_b = \texttt{VecShuffle}(\mathbb{I}'_b)$ and $\mathbb{P''}_a = \texttt{VecShuffle}(\widetilde{\mathbb{P}'_a})$ with the same order.

After performing this process, U_b finishes the blind transformation of $\mathbb{P}_a$ and $\mathbb{I}_b$. In the same time, U_b also encrypts his profile and U_a follows the same strategy to make a blind transformation towards $\mathbb{P}_b$ and $\mathbb{I}_a$.

Note that, among the above four operations, VecAdd and VecShuffle are used to conceal the original value of $\mathbb{P}_a$ and prevent U_b from obtaining the transformation ways of U_a by linking $\mathbb{P}_a$ and $\mathbb{P''}_1$. U_a (or U_b) can still obtain the correct number of matched interests and profiles since $\mathbb{P}_a$ and $\mathbb{I}_b$ (or $\mathbb{P}_b$ and $\mathbb{I}_a$) follow the same transformation pattern.

However, if only with VecAdd and VecShuffle, a dishonest participant could still infer another party's profile information without reveal his own profile information by stopping the protocol as long as he receives the matching information between his interest and another party's profile, which is called as runaway attack. Runaway attack will lead to serious unfairness issue. To achieve fairness of the proposed protocol, we further introduce VecExt and VecRev, which are used to hide the exact interest/profile matching numbers. In particular, on U_a side, VecExt introduces extra l_b ones to original matching result while VecRev introduces k_b mismatching. Therefore, the actual matching result is updated to $s_b = e_b + l_b - k_b$ for U_b and $s_a = e_a + l_a - k_a$ for U_a. The blind transformation phase is summarized in Algorithm 1.

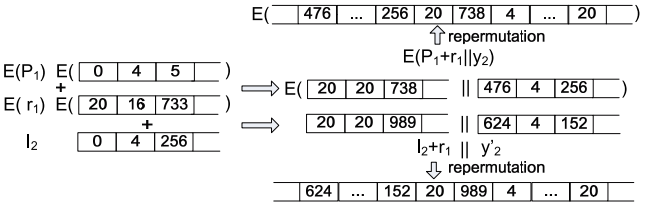


FIGURE 3. An example of profile/interest blind transformation algorithm.

In Fig. 3, we illustrate the proposed blind transformation phase by using a simple example, in which U_a's profile $\mathbb{P}_a$ and U_b's interest $\mathbb{I}_b$ are compared. To prevent the privacy leaking of $\mathbb{P}_a$ and $\mathbb{I}_b$, $\mathbb{P}_a$ and $\mathbb{I}_b$ are encrypted firstly and then are added with a randomly generated vector r_a. Since both of $\mathbb{P}_a$ and $\mathbb{I}_b$ are encrypted with Paillier cryptosystem, the homomorphic property guarantees that the comparison result will not be changed after adding the same r_a. After that,

$\mathbb{P}_a$ and $\mathbb{I}_b$ are extended and shuffled by following the same way. It is obvious that, such a transformation will not change the matching results.

D. THE PROPOSED FAIRNESS-AWARE AND COLLUSION-FREE MATCHING PROTOCOL

To check if two parties' interests match their counterparts' profiles, we have the following verification protocol. Firstly, U_a obtains $\hat{\mathbb{P}}_a = \texttt{Decrypt}(\mathbb{P}''_a, sk_a)$ by performing the decryption operation with his own secret key sk_a. After obtaining $\hat{\mathbb{P}}_a$, U_a compares it with U_b's blinded interests $\mathbb{I}_b$ to get the number of matched entries $\hat{s_b}$, while U_b could get $\hat{s_a}$ similarly. To verify if their interests and the profiles match or not, U_a sends $h_a = H(s_a||\hat{s_b})$ whereas U_b sends $h_b = H(\hat{s_a}||s_b)$ to a randomly chosen verifier. The verifier could verify whether $h_a = h_b$. If $h_a = h_b$, the match succeeds, otherwise, it fails.

A potential weakness of the proposed basic protocol is that it may be vulnerable to collusion attack. Suppose U_a and a verifier V collude, when V receives U_b's comparing result $H(\hat{s_a}||s_b)$, he sends it to U_a. Since $\hat{s_a} \leq (n + l_a)$, $s_b \leq (n + l_b)$ and n, l_a and l_b are all limited due to the efficiency of encryption and decryption overhead, U_a could get s_b by brute-force search in the complexity of $O((n + l_a)(n + l_b))$. If U_a could get h_b, based on the information he obtains, he could find out how many attributes matches in $\mathbb{P}_b$ and $\mathbb{I}_a$. Then he could figure out U_b's privacy by a probability that can't be neglected.

To thwart the collusion attack, we propose Fairness-aware and Collusion-free Matching protocol to tolerate the collusion attack based on Blind Linear Transformation [15]. The basic idea is that, instead of directly sending h_a and h_b to the verifier, an additional blind linear transformation round is introduced to protect the hash result, which is presented as follows.

1) U_a concatenates s_a and $\hat{s_b}$ to get a number $sr_a = s_a||\hat{s_b}$. He then sends $\texttt{Encrypt}(sr_a, pk_a)$ to U_b.
2) U_b generates a pair of random numbers (a_a, k_b). He then computes $\texttt{Encrypt}(sr'_a, pk_a) = \texttt{Encrypt}(a_a sr_a + k_b, pk_a)$. He also gets $sr_b = \hat{s_a}||s_b$ and transforms sr_b in the same way to obtain $sr'_b = a_b sr_b + k_b$.
3) U_b sends $\texttt{Encrypt}(sr'_b, pk_b)$ and $\texttt{Encrypt}[sr'_a, pk_a]$ back to U_a.
4) As in 2), U_a selects a pair of random numbers (a_a, k_a) and computes $\texttt{Encrypt}(sr''_b, pk_b) = \texttt{Encrypt}(a_a sr_b + k_a, pk_b)$. Decrypting with sk_a, he obtains sr'_a. He gets $sr''_a = a_a sr'_a + k_a$ and then sends $\texttt{Encrypt}(sr''_b, pk_b)$ back to U_b.
5) U_b decrypts $\texttt{Encrypt}(sr''_b, pk_b)$. U_a sends $H(sr''_a)$ and U_b sends $H(sr''_b)$ to verifier to test their equality.

In this way, both of their hash results are preserved by a pair of blinding numbers which are much larger than $(n + l_a)$ or $(n + l_b)$, thus collusion attack is considered impossible under this scheme for the expensive computation cost.

IV. SECURITY ANALYSIS

In this section, we will demonstrate the fairness and the privacy of the proposed protocol by the detailed security analysis.

A. SECURITY AGAINST INTEREST/PROFILE LEAKING

Without loss of generality, we just consider $\mathbb{P}_a$ and $\mathbb{I}_b$. Since the profile $\mathbb{P}_a$ is encrypted by Pallier Cryptogsystem, and without the secret key sk_a, no one except U_a could get $\mathbb{P}_a$. Thus the privacy in $\mathbb{P}$ could be preserved.

The privacy in interest $\mathbb{I}_b$ is guaranteed by BPVT protocol. Since after receiving the processed $\mathbb{P}_a$ and $\mathbb{I}_b$, U_a can not correlate any item of $\mathbb{I}_b$ with the attributes in $\mathbb{P}_a$. At the same time, it is guaranteed for U_b that U_a can not test his interest by changing $\mathbb{P}_a$ arbitrarily.

B. SECURITY AGAINST RUNAWAY ATTACK

As we have introduced in Section III, after U_a decrypts the processed $\mathbb{P}_a$, he could obtain the comparison result $\hat{s_b}$ which indicates how many pairs of items between processed $\mathbb{P}_a$ and $\mathbb{I}_b$ match. If he knows $m_b = l_b - k_b$ which indicates how many pairs are the same in appended vector, he will knows that there are $s_b - m_b$ pairs of attributes matched between $\mathbb{P}_a$ and $\mathbb{I}_b$. Therefore, U_a could randomly select $s_b - m_b$ attributes in his profile $\mathbb{P}_a$ as the corresponding attributes in $\mathbb{I}_b$, if this probability could not be neglected, U_a may abort the protocol and get some of U_b's interest with a high probability.

We will use the following Theorem to discuss the upper bound of the successful probability that U_a could guess any item of $\mathbb{I}_b$ without any error.

Theorem 1: Given a profile $\mathbb{P}$ and an interest $\mathbb{I}$ which are blind transformed and matched by following the proposed protocols, the correct-guess probability $P(CG)$ that U could infer any item of $\mathbb{I}$ based on the blind transformed $\mathbb{P}$ and the comparing result s is bounded by $\frac{3}{ln}(n \geq 5)$, where n is the length of $\mathbb{P}$ and l is the number of attributes appended to $\mathbb{P}$.

Proof: The successful guess probability is expressed as:

$$P(CG) = \sum_{m'=1}^{\min(s,l)} p(m = m') Pr\{CG|s, m\} \tag{1}$$

where $p(m = m')$ is the probability that U_1 could guess m correctly, and in our scheme, $p(m = m') = \frac{1}{l}, m' \in \{1, 2, \ldots, l\}$. $Pr\{CG|s, m\}$ is the probability that given s and m, U_1 could guess $s - m$ items correctly. Obviously, when $s - m \geq n$, $P(CG|s, m) = 0$, when $s - m < n$, $P(CG|s, m) = \frac{1}{\binom{n}{s-m}}$. No matter $s > l$ or $s \leq l$, we could get

$$P(CG) = \frac{1}{l} \sum_{m=1}^{s-1} \frac{1}{\binom{n}{s-m}} \tag{2}$$

By mathematical induction, $P(CG) \leq \frac{3}{ln}(n \geq 5)$. □

Theorem 1 indicates that given ϵ as the expected secure probability such that $P(CG) < \epsilon$, if ϵ is small enough, then we think U_1 will get nothing about U_2's interest since he could not guess any part of $\mathbb{I}_2$ correctly, thus he has no incentive to

abort the protocol. Furthermore, we could safely bound it with $\frac{3}{ln} < \epsilon$. And according to this inequality, we could calculate l and m to guarantee the fairness. Theorem 1 also indicates that if two users are not matched finally, they could not guess anything according to the comparing result. Thus the fairness is guaranteed by the proposed protocol.

C. SECURITY AGAINST COLLUSION ATTACK

In the revealing phase, the number of matches on both sides will be transformed and neither side knows how the other side performs such transformation. Obviously, the probability of guessing $(a_i, b_i), i \in (1, 2)$ of the other side is negligible. We have the following theorem.

Theorem 2: Given $H(sr''_b)$, the probability of guessing $\hat{s_a}$ and s_b correctly is negligible.

Proof: The attempt to guess the parameters can be formalized as guessing (a, b) in $y = ax + b$ given the knowledge of only one pair of (x, y), which is negligible. With a and b, the result x is transformed into a larger space, making exhaustive enumeration difficult. Thus, this step prevents either side from guessing the actual value of the other side by brute-force search over the hash value. In other words, assuming one-way characteristic of the hash function, both users have no knowledge of the other side's query results. □

The Fairness Assurance in matching phase is achieved in that the only results revealed so far are "success" or "fail", which is known to both sides at the same time. The verifier only receives two hash values and should only answer whether they are equal or not. The only information available to them are the two hash values and the only answer they can get is whether they are equal, which is just as what we intended. No further information can be derived from the hash value due to one-way characteristic. Thus, the revealing phase reveals information no more than "success" or "fail".

V. EVALUATION

We implemented our protocol in Java for portability and evaluated it on a laptop with Intel Core i3-330m (2.1 GHz) and 2 GB RAM. The Paillier encryption library was based upon [16]. We modified it and used the fast variant of Paillier scheme as proposed in [14]. We evaluated the running time of our protocol in Blind Transformation, Fair Matching and Blind Linear Transformation phase. The algorithm used in Blind Shuffle is Knuth Shuffle [17]. We use it in order to guarantee the randomness in permutation.

The Paillier Key length is selected as 1024-bit. The α is 160-bit as in [14]. We tested a single round in our simulation in which U_1 launches the protocol and matches his profile against U_2's interest. As in a real world application such protocol will be executed in parallel by two users, a single round is enough to measure its efficiency. The results are depicted in Figs. 4–7. We choose 3 security parameter length $l = n$, $l = 2n$, $l = 3n$. The number of attributes range from 20 to 100. We measure the running time against the number of attributes under those 3 parameter settings. We've plotted the average value of

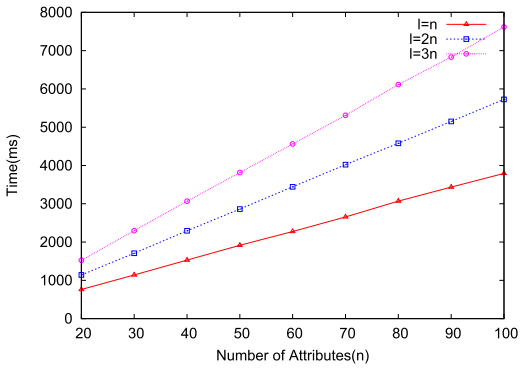


FIGURE 4. Execution time on blind transformation for different number of attributes (ms).

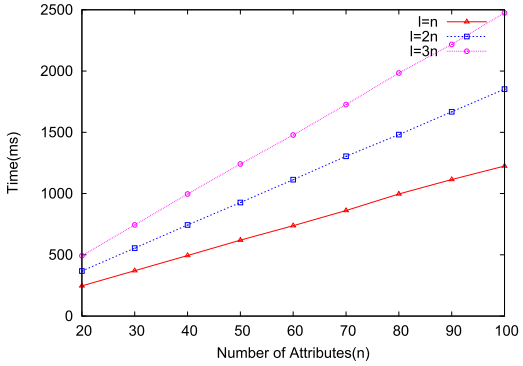


FIGURE 5. Execution time on fair matching phase for different number of attributes (ms).

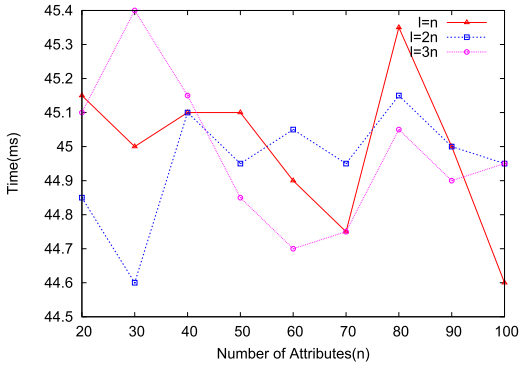


FIGURE 6. Execution time on blind linear transformation for different number of attributes (ms).

20 runs. Detailed statistics about the results is shown in Table 2.

As we can see from Figs. 4–7, the growth of the execution time remains linear almost in all cases. This makes sense since the time taken in each encryption or decryption is relatively constant as the key size is fixed. Thus the decryption or encryption time grows in proportion with the number of total attributes.

Fig. 4 also indicates that, most of the computation time is spent in Blind Transformation phase. This is true as the encryption is the most expensive part in our implementation.

TABLE 2. Statistics of experiment results.

n	l	Blind Transformation			Fair Matching			Blind Linear Transformation			Total		
		Min	Max	Std	Min	Max	Std	Min	Max	Std	Min	Max	Std
20	20	753	784	12.3333	240	257	41.4620	44	47	0.7263	1038	1088	16.9773
20	40	1129	1177	17.3433	361	382	5.9521	44	46	0.7263	1536	1603	23.1471
20	60	1505	1567	24.9469	480	514	10.0668	44	48	1.1790	2034	2127	34.7969
60	60	2257	2352	31.3503	717	767	11.6645	44	47	0.7681	3018	3162	42.5677
60	120	3389	3550	63.3628	1079	1159	23.8778	44	47	1.1169	4515	4756	86.4866
60	180	4516	4729	72.3795	1446	1527	24.3234	44	47	0.9000	6009	6294	94.7878
100	100	3765	3918	52.0802	1204	1273	16.0409	44	46	0.7348	5016	5234	65.2713
100	200	5647	5914	101.2305	1786	1930	40.1426	44	47	0.8646	7477	7882	139.7714
100	300	7526	7841	125.4932	2395	2588	53.3105	44	46	0.7399	9965	10464	175.8158

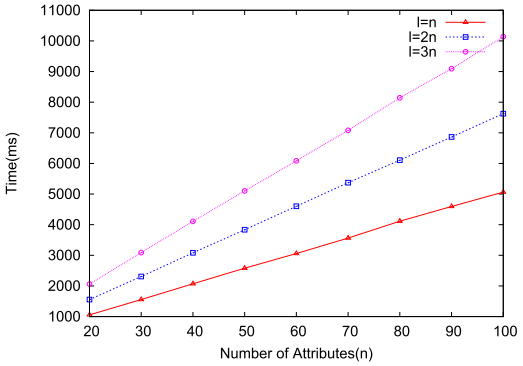


FIGURE 7. Total execution time required vs number for different number of attributes (ms).

As Fig. 5 shows, the Blind Linear Transformation phase in the protocol introduces quite low overhead, within 46ms in the transformation step. Thus, compared with the basic scheme, the advanced scheme is secure yet runs with little overhead.

Using different security parameter l will give different performance since l increases the total vector size and the number of encryption/decryption operations. We present these 3 parameter settings to demonstrate a trade-off between security and efficiency. However, given the fact that even if the adversary has guessed one attribute correctly, he has no way to verify it and thus, setting $l = n$ is enough in most cases since in comparison with the number of attributes (normally 20 to 30 as in [5], [6]), the $\frac{3}{ln} = \frac{3}{n^2}(l = n)$ is far less than a random guess with probability $\frac{1}{n}$. Thus it's secure enough in most cases. With $l = n$, our implementation performs [6] with 40% less running time (1.556s compared to 2.6s in [6]). As [6] runs on Intel Core Duo P8600 (2.4GHz), whose clock speed is faster than our i3-330m, and the simulation is a single thread task with no speed up provided by the Hyper-Threading technology in Core i3-330m, we can safely conclude that our scheme is relatively efficient.

Note that the computation overhead on third party users is not measured in the simulation. But it's clear that the only task for third party users is to test whether two integers(no larger than 256 bit when using SHA-256) are equal. The transformation overhead and power consumption for them is negligible.

VI. RELATED WORK

Our work is related to the following previous works.

A. MOBILE SOCIAL NETWORKS

The explosive popularity of online social networks has attracted significant attention recently [18], [19]. In [20], social serendipity to perform matchmaking in mobile social networks is presented. In [21], Loopt is a mobile geo-location service that notifies users of friends' location and activities via detailed interactive maps. It is also observed that there is a large body of industrial efforts, which try to make location based friend discovery by providing andriod or IOS based services. For example, WeChat is a popular mobile social network app which provides "Look Around" function [22]. With this function, the mobile users could review the profiles of other mobile users who are physically nearby and then communicate with interested users. Other typical apps include Skout [23], Momo [24] and others. However, most of these existing apps fail to consider hide users' profiles. Therefore, designing a privacy-preserving friend matching protocol is highly desired for these apps.

B. SECURE FRIEND DISCOVERY IN MOBILE SOCIAL NETWORKS

Dong *et al.* proposed to match two PMSN users based on the distance between their social coordinates in an online social network [6]. In [5], Li *et al.* proposed FindU, a privacy-preserving personal profile matching in mobile social networks. By using secure multi-party computation (SMC) techniques, it can achieves that, an initiating user can find from a group of users the one whose profile best matches with his/her. In [25] and [26], it proposed the concept of Fine-Grained Private Matching, which allows finer differentiation between PMSN users and can support a wide range of matching metrics at different privacy levels. Different from these existing works, we separate users' profiles from their interest for the first time. Further, we propose a novel Run-away attack, which may potentially introduce the unfairness issue. The proposed scheme could well thwart this novel attack and thus achieve a better security.

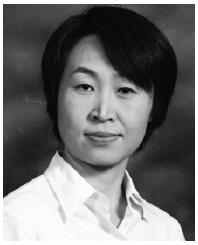
VII. CONCLUSION

In this work, we have developed a novel protocol that will ensure the fairness and the privacy of privacy-preserving interest and profile matching process in mobile social networks. Our future work includes how to provide fine-grained interest/profile matching and investigate more security and privacy issues in mobile social networks.

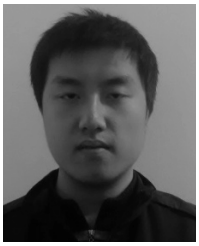
REFERENCES

[1] (2012). *Foursquare* [Online]. Available: https://foursquare.com/
[2] (2012). *Gowalla* [Online]. Available: http://gowalla.com/
[3] Z. Yang, B. Zhang, J. Dai, A. C. Champion, D. Xuan, and D. Li, "ESmallTalker: A distributed mobile system for social networking in physical proximity," in *Proc. IEEE ICDCS*, Jun. 2010, pp. 468–477.
[4] R. Shokri, G. Theodorakopoulos, C. Troncoso, J.-P. Hubaux, and J.-Y. Le Boudec, "Protecting location privacy: Optimal strategy against localization attacks," in *Proc. ACM Conf. CCS*, 2012, pp. 617–627.
[5] M. Li, N. Cao, S. Yu, and W. Lou, "FindU: Privacy-preserving personal profile matching in mobile social networks," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 2435–2443.
[6] W. Dong, V. Dave, L. Qiu, and Y. Zhang, "Secure friend discovery in mobile social networks," in *Proc. IEEE INFOCOM*, Shanghai, China, Apr. 2011, pp. 1647–1655.
[7] R. Lu, X. Lin, X. Liang, and X. Shen, "A secure handshake scheme with symptoms-matching for mhealthcare social network," *Mobile Netw. Appl.*, vol. 16, no. 6, pp. 683–694, 2010.
[8] M. Von Arb, M. Bader, M. Kuhn, and R. Wattenhofer, "VENETA: Serverless friend-of-friend detection in mobile social networking," in *Proc. IEEE WIMOB*, Oct. 2008, pp. 184–189.
[9] L. Kissner and D. Song, "Privacy-preserving set operations," in *Advances in Cryptology—CRYPTO 2005*, pp. 241–257.
[10] Q. Ye, H. Wang, and J. Pieprzyk, "Distributed private matching and set operations," in *Proc. Information Security Practice and Experience (ISPEC) Conf.*, 2008, pp. 347–360.
[11] M. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology—EUROCRYPT 2004*, pp. 1–19.
[12] E. D. Cristofaro and G. Tsudik, "Practical private set intersection protocols with linear computational and bandwidth complexity," in *Financial Cryptography and Data Security*. Berlin, Germany: Springer-Verlag, 2010, pp. 143–159.
[13] (2012). *Perfect-Match* [Online]. Available: http://www.perfectmatch.com
[14] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology—EUROCRYPT 1999*, pp. 223–238.
[15] R. Rivest, "Unconditionally secure commitment and oblivious transfer schemes using private channels and a trusted initializer," unpublished.
[16] (2012). *Paillier Homomorphic Cryptosystem (Java Implementation)* [Online]. Available: http://www.csee.umbc.edu/kunliu1/research/Paillier.html
[17] D. E. Knuth, *The Art of Computer Programming, Volume 2: Seminumerical Algorithms*. Reading, MA, USA: Addison-Wesley, 1997.
[18] H. Zhu, X. Lin, R. Lu, Y. Fan, and X. Shen,"SMART: A secure multilayer credit-Based incentive scheme for delay-tolerant networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 8, pp. 828–836, Oct. 2009.
[19] H. Zhu, X. Lin, R. Lu, P.-H. Ho, and X. Shen, "SLAB: Secure localized authentication and billing scheme for wireless mesh networks," *IEEE Trans. Wireless Commun.*, vol. 17, no. 10, pp. 3858–3868, Oct. 2008.
[20] N. Eagle and A. Pentland, "Social serendipity: Mobilizing social software," *IEEE Pervas. Comput.*, vol. 4, no. 2, pp. 28–34, Apr. 2005.
[21] *Loopt* [Online]. Available: http://www.loopt.com Accessed: 4 Apr. 2013
[22] *Wechat* [Online]. Available: http://www.wechat.com/ Accessed: 4 Apr. 2013
[23] *Skout* [Online]. Available: http://www.skout.com/ Accessed: 4 Apr. 2013
[24] *Momo* [Online]. Available: http://immomo.com/ Accessed: 4 Apr. 2013
[25] R. Zhang, Y. Zhang, J. Sun, and G. Yan, "Fine-grained private matching for proximity-based mobile social networking," in *Proc. IEEE Int. Conf. Comput. Commun.*, Orlando, FL, USA, Mar. 2012, pp. 1969–1977.
[26] R. Zhang, J. Zhang, Y. Zhang, J. Sun, and G. Yan, "Privacy-preserving profile matching for proximity-based mobile social networking," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 9, pp. 656–668, Sep. 2013.

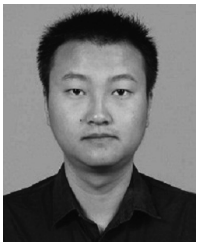
HAOJIN ZHU (M'09) received his B.Sc. degree from Wuhan University, Wuhan, China, in 2002, his M.Sc. degree from Shanghai Jiao Tong University, Shanghai, China, in 2005, both in computer science, and the Ph.D. degree in electrical and computer engineering from the University of Waterloo, Waterloo, ON, Canada, in 2009. He is currently an Associate Professor with the Department of Computer Science and Engineering, Shanghai Jiao Tong University. His current research interests include wireless network security and distributed system security. He is a co-recipient of Best Paper Awards from the IEEE ICC - Computer and Communications Security Symposium in 2007 and Chinacom-Wireless Communication Symposium in 2008. He served as a Guest Editor for the *IEEE Networks* and an Associate Editor for *KSII Transactions on Internet and Information Systems* and *Ad Hoc & Sensor Wireless Networks*. He currently serves as the Technical Program Committee for international conferences such as INFOCOM, GLOBECOM, ICC, and WCNC.



SUGUO DU received the B.Sc. degree in applied mathematics from the Ocean University of Qingdao, Qingdao, China, in 1993, the M.Sc. degree in mathematics from Nanyang Technological University, Singapore, in 1998, and the Ph.D. degree from the Control Theory and Applications Centre, Coventry University, Coventry, U.K., in 2002. She is currently an Associate Professor with the Management Science Department, Antai College of Economics and Management, Shanghai Jiao Tong University, Shanghai, China. Her current research interests include risk and reliability assessment, fault tree analysis using binary decision diagrams, fault detection for nonlinear system, and wireless network security management.



MUYUAN LI (S'12) received his B.Sc. degree in computer science from Shanghai Jiao Tong, Shanghai, China, in 2013. His research interests include security and privacy in mobile social networks.



ZHAOYU GAO (S'12) received his B.Sc. degree in applied mathematics from Wuhan University, Wuhan, China, in 2009, and the M.Sc. degree from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2013. His research interests include cognitive radio network, security and privacy of wireless network.