Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2016, Article ID 8575187, 12 pages
http://dx.doi.org/10.1155/2016/8575187

Research Article

PRUB: A Privacy Protection Friend Recommendation System Based on User Behavior

Wei Jiang, Ruijin Wang, Zhiyuan Xu, Yaodong Huang, Shuo Chang, and Zhiguang Qin

University of Electronic Science and Technology of China, Chengdu 611731, China

Correspondence should be addressed to Ruijin Wang; 124355850@qq.com

Received 13 February 2016; Accepted 2 August 2016

Academic Editor: Salvatore Alfonzetti

The fast developing social network is a double-edged sword. It remains a serious problem to provide users with excellent mobile social network services as well as protecting privacy data. Most popular social applications utilize behavior of users to build connection with people having similar behavior, thus improving user experience. However, many users do not want to share their certain behavioral information to the recommendation system. In this paper, we aim to design a secure friend recommendation system based on the user behavior, called PRUB. The system proposed aims at achieving fine-grained recommendation to friends who share some same characteristics without exposing the actual user behavior. We utilized the anonymous data from a Chinese ISP, which records the user browsing behavior, for 3 months to test our system. The experiment result shows that our system can achieve a remarkable recommendation goal and, at the same time, protect the privacy of the user behavior information.

1. Introduction

We are now embracing the era of the mobile social network. This network connects people with similar interests and characteristics through mobile devices like smartphones, tablets, and so forth. Users on a mobile social networks platform can share their states conveniently. The mobile social network platform is an open platform: it is established on the base of actual social relationship and developed by further expanding the social circle. Aiming at recommending new links for each user, almost all the mobile social networking systems provide a service to recommend friends. Considering the fact that the size of the network grows exponentially, many service providers and researchers are trying to import distributed management to implement a recommendation system as a way to ease the pressure of services on a centralized management system [1] and improve user experience at the same time.

The recommendation system can achieve a promising result using the information of user behavior [2]. However, some of the information contains personal privacy data, and users are unwilling to share their certain behavioral information to others. They prefer to exchange their information with people who share common interests, instead of all the strangers. Particularly in the distributed systems, because there are no regulators in the interactive process, the privacy security will be a problem. Thus, "how to realize high quality recommendation as well as protecting the privacy of user behavior information of the user" is becoming a research hotspot [3–5].

To solve these problems, we design a secure friend recommendation system based on the user behavior, called PRUB. This system aims at achieving fine-grained recommendation to friends who share common interests without exposing the actual user behavior. PRUB provides a modified matching protocol and authorization protocol to ensure the security of user behavior information in the hybrid management which combines the centralized and distributed management.

PRUB works mainly in two steps. The first step is to realize the coarse friend recommendation. The authentication server classifies users using the KNN classification algorithm which is based on user behavior information, such as users' browsing records, and returns friend recommendation result based on people who share the same interests as the users. The second step is to realize the fine-grained friend recommendation. Each user uses coarse grained friend recommendation on similarity calculation using the matching protocol. If the similarity degree is greater than the user defined threshold,

PRUB adds the person to the fine-grained friend recommendation list.

This paper aims to make the following contributions:

(1) We present a hybrid management architecture according to the mobile social networks platform characteristics, easing the pressure from the server and improving the user experience.

(2) We propose a privacy supported matching protocol. Utilizing the users' behavior, it can realize the personalized instead of the blindly recommended result. At the same time, the protocol ensures personal privacy security; users can protect their own sensitive information and avoid exposing it to all the strangers on the platform.

(3) We define a security model and theoretically analyze the security of our protocol. We aim to prove that our protocol can defend against attack in the position of initiator and matching target, respectively.

To evaluate our system PRUB, we utilize anonymous data from a Chinese ISP, which records the user browsing behavior for 3 months. The experiment result shows that our system can achieve a fine-grained recommendation and protect the privacy of the user behavior information at the same time.

The rest of the paper is organized as follows. Section 2 discusses related works. Section 3 provides the system overview of PRUB. The secure matching protocol is discussed in Section 4. Section 5 analyzes the security of the system. The experiment result is presented in Section 6. We conclude our work in Section 7.

2. Related Works

Corresponding to the structure of the mobile social network, there are three application patterns for friend recommendation: centralized management, distributed management, and hybrid management.

Centralized Management. In this pattern, all the user information is stored on the central server. As a trusted third party, the central server manages the whole system and handles all the processes. Users only need to access the server using a mobile client and they can acquire the desired service. The central server will complete the characters matching and return the recommended friend list to the users [8]. Throughout the process, there is no interaction between the users. In the centralized management, the central server holds all the information about user characteristics. It can effectively protect the user privacy through enhancing the server security. However, the server cannot be accessed at any time; this depends on the network conditions. Thus, user experience decreases under unstable network condition. Apart from this, not all the users are willing to deliver their behavior information to server providers, especially some personal privacy information. Some server providers may utilize the information for illegal activities [9].

Distributed Management. In distributed social networks, data are stored and handled at the local clients, and users can directly interact with each other. Clients broadcast their own information and receive information of others at the same time and then match characteristics based on the information to discover target users. In this pattern, the whole recommendation process can be realized among clients without any server participation [10]. This pattern certainly relieves the pressure of the server. However, clients may unintentionally publish some unnecessary information, even user privacy [11]. In the distributed system, the interaction process lacks control and fails to ensure the security of the recommendation process. Many researchers presented secure matching protocols to avoid the shortage, such as [12]; it utilizes Shamir secret dividing to complete matching.

Hybrid Management. In order to integrate the advantages of centralized and distributed management, some researchers presented hybrid management [13]. In this pattern, users could interact directly, but, during the process of matching, appropriate server control is required, such as storing temporary data and arbitration. The hybrid management pattern can reduce the server stress and provide security mechanism as well. The problem for researchers is how to minimize the information provided by users, how to realize secure matching with server involvement as little as possible, and how to realize arbitration using minimal user information and ensure accuracy [14].

Our recommendation system PRUB is built on the hybrid management. Matching protocols is the core of hybrid management; users can find some friends who share common interests, such as common browsing habits, and also protect their private information. The process of matching can be regarded as PSI (private set intersection) problem or PCSI (private cardinality of set intersection) problem [15]. Currently, the popular resolution algorithm can be categorized into three types.

Matching Protocol Based on Commutative Encryption Function. Agrawal et al. [7] presented a commutative encryption protocol to solve PSI/PCSI problem. It used a pair of encryption functions f and g and $f(g(x)) = g(f(x))$; the property of each function is that the encrypted result is independent of the calculation order, such as $f_e(x) = x^e \bmod p$, where p is safe prime. This protocol is secure under the hypothesis of DDH (Decisional Diffie-Hellman) hypothesis, and only one of the participators knows the intersection; the other cannot acquire anything. However, this protocol cannot defend against malicious attacks.

Von Arb et al. [16] presented a social platform VENETA based on the algorithm of Agrawal et al. They rely on a construction based on commutative encryption. Compared to Agrawal et al., they assume that the attack cannot cause any serious damage. A victim only reveals a contact he was willing to share, without getting this information in return. VENETA enables two users to calculate the intersection of their characters in a certain range. If it successfully matches between two users, VENETA will recommend this stranger to the user.

Xie and Hengartner [6] presented a matching protocol in mobile social networks. The protocol adds signature

verification to property elements; this paper shows that Agrawal et al. proved that, given the Decisional Diffie-Hellman (DDH) hypothesis, $\langle X_i, f_e(X_i), Y_j, f_e(Y_j)\rangle$ for fixed values of i and j, with $f_e(x) = x^e$, is indistinguishable from $\langle Xi, f_e(X_i), Y_j, Z\rangle$, when e is not given. They prove that these certificates are sent across an encrypted channel, so a passive eavesdropper cannot learn Alice's or Bob's interests. Thus, attackers cannot reorder all the segments and properties easily, thus avoiding counterfeit and scanning attacks.

In [17], a new efficient solution to Yao's millionaires' problem based on symmetric cryptography is constructed, and the privacy-preserving property of the solution is demonstrated by a well-accepted simulation paradigm. It proposes a new security paradigm that quantitatively captures the security levels of different solutions. The paper proposes an ideal model with a trusted third party; Alice and Bob have x and y; they want privately to compute functionality $f(x, y) = (f_1(x, y), f_2(x, y))$ with the help of a trusted third party. At the end of the protocol, Alice (Bob) obtains $f_1(x, y)$, $f_2(x, y)$ without leaking $x(y)$. The ideal model provides the best possible security of secure multiparty computations, and its security level is the highest level compared with that any secure multiparty computation solution can achieve. The authors of [17] use the following to judge whether Protocol π_1 is more secure than Protocol π_2:

$$\frac{H_{\pi_1}(x \mid f_2(x, y))}{H_{\pi_2}(x \mid f_2(x, y))} > 1,$$
$$\text{or } \frac{H_{\pi_1}(y \mid f_1(x, y))}{H_{\pi_2}(y \mid f_1(x, y))} > 1, \quad (1)$$
$$\text{or both hold.}$$

They concluded that the new solution was as secure as both the ideal secure multiparty computation solution and Yao's solution. In this paper, the solution provided in [17] enables XOR operating in commutative encryption function, but it greatly increases the calculation costs and decreases the security of the system.

Matching Protocol Based on Linear Polynomial. The paper [15] presented FNP protocol, which transforms properties into linear polynomial, and uses the character of homomorphy to handle the encrypted coefficients. In the exchange process, one side is the client and the other is the server. For each input property of the client, it only knows whether the property belongs to the server and cannot acquire any other information. Meanwhile, the server cannot get any other input information from the client.

Kissner and Song [18] used a polynomial to represent multiple collections. Taking advantage of polynomial and addition homomorphy function, it realizes the secure operation of intersection, union, and complementation. This protocol can be applied against semihonest attacks and these techniques can be applied to a wide range of practical problems. This paper has an important feature of privacy-preserving multiset operations which can be composed and enable a wide range of applications. The authors of [18] use the following grammar to compute the output of any function over the multisets.

$Y ::= s|\text{Rdd}(Y)|\ Y \cap Y|s \cup Y|Y \cup s$. They construct an algorithm for computing the polynomial representation of operations on sets, including union, intersection, and element reduction. And they extend these techniques including intersection and element reduction with a trusted third party to encrypted polynomials, allowing secure implementation of our techniques without a trusted third party. Dachman-Soled et al. [19] presented a PSI protocol which can be applied for defending against malicious users. They also use the polynomial coefficient to represent properties and use Shamir Secret Share to dividing coefficient for realizing higher security.

Lu et al. [20] put forward Secure Handshake with symptom-matching and deployed the matching protocol into disease monitoring. Their system enables patients who have the same symptoms to communicate and share information. The core of their algorithm is the feature of bilinear matching function. In this system model, they consider a typical mHealthcare social network (MHSN), which consists of trusted authority (TA) at eHealth center and a large number of mobile patients. As the patient health condition is very sensitive to the patient himself/herself, therefore, it is essential that the privacy of PHI should be controlled by the patient in a MHSN environment, so they develop a secure same-symptom-based handshake (SSH) scheme. It consists of these algorithms: system setup, patient joining, and patient's same-symptom-based handshaking (PatientsSSH). The paper shows the employed identity-based encryption.

$$C = \begin{cases} C_1 = r\left(P_{\text{pub}} + H(\text{pid})P\right), & \text{where } r \xleftarrow{R} Z_q^* \\ C_2 = e\left(P, H_0(T)\right)r \cdot N, & T \text{ is specific triage.} \end{cases} \quad (2)$$

The identity-based encryption (IBE) should be semantic security (indistinguishable) under selective-PID-symptoms and chosen-plaintext attacks. In this paper, let IBE be a secure encryption scheme with security parameter l, and define the advantage probability of A to be an INDsPS-CPA adversary against IBE. It is very important in the random oracle model.

$$\begin{aligned} \text{Adv}_{\text{IBE},A}^{\text{IND-sPS-CPA}}(l) &= 2 \cdot \Pr\left[\text{Exp}_{\text{IBE},A}^{\text{IND-sPS-CPA}}(l) = 1\right] - 1 \\ &= 2 \cdot \Pr\left[b = b'\right] - 1. \end{aligned} \quad (3)$$

In this paper, SSH is of vital importance to the success of MHSN, but this algorithm can be only applied to matching single property and it is difficult to extend it into multiproperties.

Matching Protocol Based on Pseudorandom Number. This protocol is firstly presented in [21]. Hazay and Lindell designed a PSI protocol, in order to defend against different attacks and ensure the operating efficiency at the same time. The pseudorandom number is used for encryption. In this paper, the protocols for securely computing the set intersection functionality are based on secure pseudorandom function evaluations, in contrast to previous protocols. This paper

proposed Secure Pattern Matching; F_{PM} was used to address the question of how to securely compute the above basic pattern matching functionality.

$$((T, m), p) \longmapsto \begin{cases} (\lambda, \{i \mid T_i = p\}) & \text{if } |p| \le m \\ (\lambda, \{i \mid T_i = p_1 \cdots p_m\}) & \text{otherwise.} \end{cases} \quad (4)$$

This protocol is presented for securely computing F_{PM} in the presence of *malicious adversaries* with *one-sided simulatability*. And specific properties of the Naor-Reingold pseudorandom function and the protocol π_{PRF} for computing the Naor-Reingold function are utilized. The following is used instead of the corrupted party p_1 *computing and sending the set*.

$$\left\{\left(i, g_i = \widetilde{g}^{\prod_{j=1}^{\log N} a_{m+j}^{(i)_j}}\right)\right\}_{i=1}^{N-m+1}. \quad (5)$$

So, the Naor-Reingold pseudorandom function will achieve high efficiency.

Yang et al. [22] designed a distributed mobile social network, E-SmallTalker. It utilizes Bloom filter as the store structure of properties and calculates the intersection through several rounds of iteration with pseudorandom function. E-SmallTalker can reduce the storage space effectively and prevent interactive users from acquiring more information besides the common properties. It requires no data services like Internet access and exchanges user information between two phones and performs matching locally. Yang et al. build on the Bluetooth Service Discovery Protocol (SDP) to search for nearby E-SmallTalker users. And the iterative Bloom filter (IBF) they proposed is to encode user information; data is encoded in a bit string to address SDP attributes' size limit. This system architecture includes four software components: context data store, context encoding and matching, context exchange, and user interface (UI). The authors of [22] devise a multiround protocol to achieve the desired false-positive rate f with a minimum total amount of transmission given the constraints imposed by the implementation of Bluetooth SDP, and the quantitative measurement of the false-positive rate is defined by the following formula: $f = (1 - (1 - 1/m)^{kn})^k \approx (1 - e^{-kn/m})^k$. So, this paper's approach was efficient in computation and communication.

3. PRUB System Overview

The PRUB system is based on users browsing behavior to recommend related friends with similar behavior and activities. The system adopts the hybrid management architecture mentioned above. The system consists of several users, several smartphones, one verification server (VS), and several anchor servers (AS). The basic structure is shown in Figure 1.

3.1. Users. Every user who wants to use our system to find friends with similar behavior should have a smartphone and install our application. The application will collect users browsing histories and classify them into several catalogs. The value of each catalog is defined as properties and characteristic of the user. The users should sign up with an account to use the service.

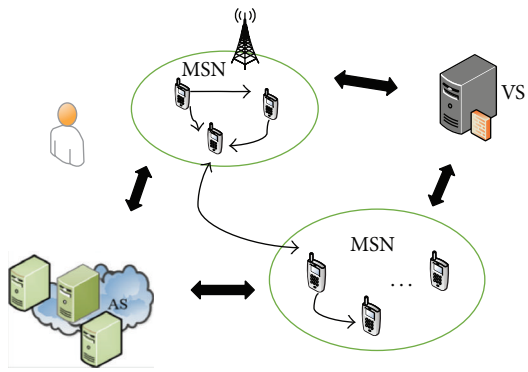


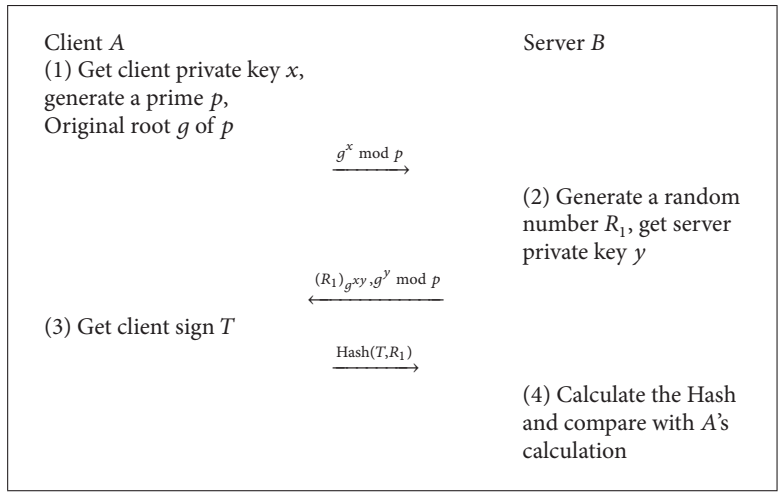
Figure 1: PRUB system structure.

To test the security of our system, we define some kinds of abnormal users. The first kind of abnormal users will not try to break protocols of the network. They only want to obtain some privacy data of other users by analyzing the information they obtain from the recommendation result. The way they are trying to do this is defined as passive attack. These users are called semihonest users. Another kind of users who are trying to attack actively is called malicious users. For getting more information of other users, they do not obey the protocols and even try to break down the whole system. Some methods of attacking are to send fake messages or terminate an agreement before it finishes.

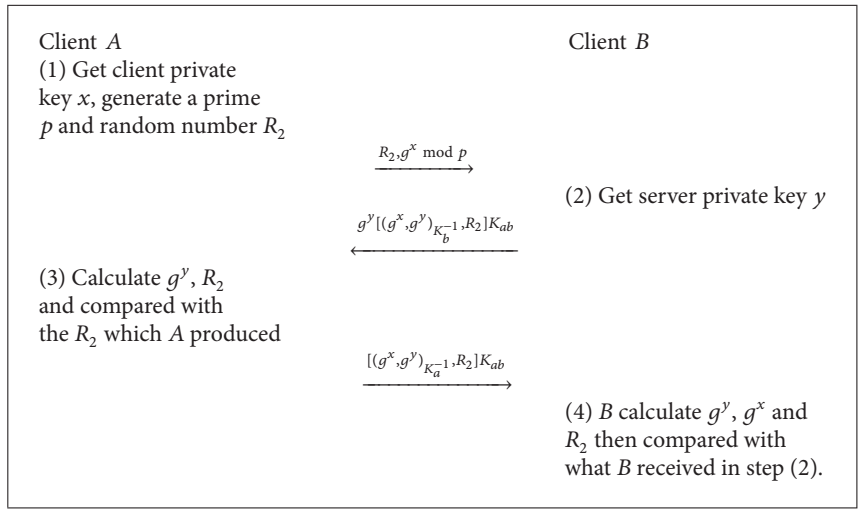
3.2. Smartphones. The smartphones which have installed our application are also part of the system. The application saves the personal information of the user, including his/her ID, properties, private key, and public key. The smartphone should have some computability in order to compute some secure information. The smartphones of the users form mobile social networks (MSNs).

3.3. Verification Server (VS). The user should register on the verification server before he/she obtains the recommendation data. The application will send the encrypted user's personal information to the VS to get the ID and a pair of RSA keys. When verifying the user's ID and properties, the application will send the username and the public key to VS; the VS generates a random number and then sends the random number to the user's application. The application will utilize the properties $(X_1, X_2, \ldots, X_m)$ to get $\mathrm{Attr} = (X_1)^a \parallel (X_2)^a \parallel \cdots \parallel (X_n)^a$ and send the ID to the VS. The VS uses its private key to get $\mathrm{sign}_{\mathrm{vs}}(\mathrm{ID} \parallel \mathrm{Attr})$ and sends it back to the user's application. In order to prevent abnormal users from changing their properties to obtain others' private information, the VS can bind the user with its properties and signature certification.

3.4. Anchor Server (AS). The anchor servers are used to connect several MSNs. The user can register on several



Box 1: One-way authentication protocol.



Box 2: Mutual authentication protocol.

different MSNs, and the AS transmits information to the users from different MSNs.

4. Secure Matching Protocol

One-Way Authentication Protocol. In the beginning of establishing the interaction communication channel between VS and the user, VS utilizes one-way authentication protocol to authenticate the identity of the user. The protocol is described in Box 1.

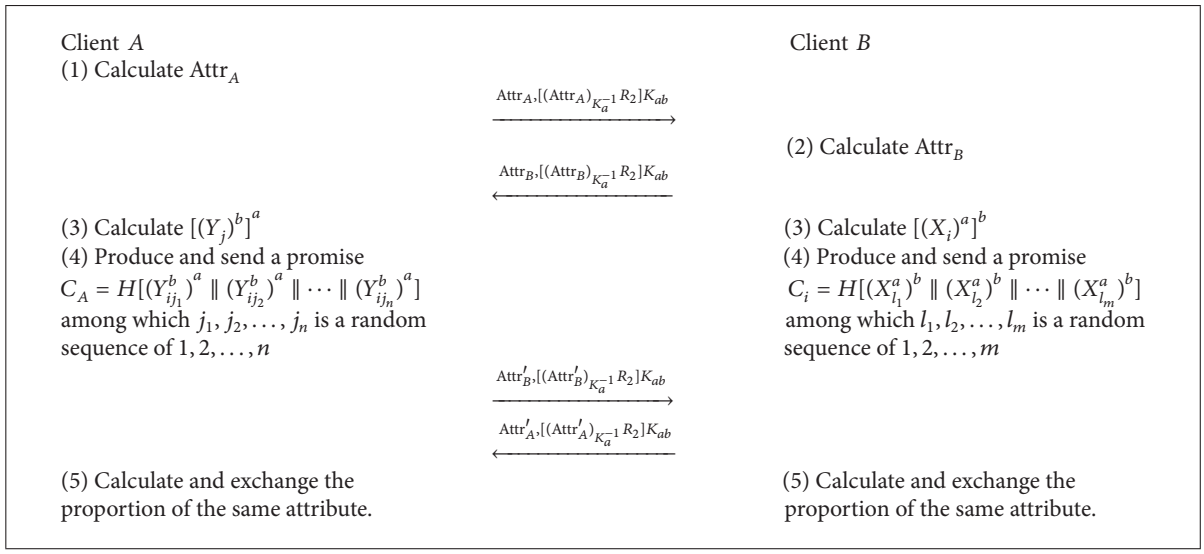
After the interaction about setting identity and random identification, the protocol can realize the identification between dual direction key and one-way entity through verifying hash value. It achieves forward secrecy and nonrepudiation and can defend the man-in-the-middle attacks, including replay attack, reflection attack, prophecy attack, and interleaving attack.

Mutual Authentication Protocol. This protocol is used to mutually authenticate the pairing of interaction when establishing the channel. The protocol is described in Box 2, where $K_A = g^x \bmod p$, $K_B = g^y \bmod p$, and $K_{AB} = g^{xy} \bmod p$.

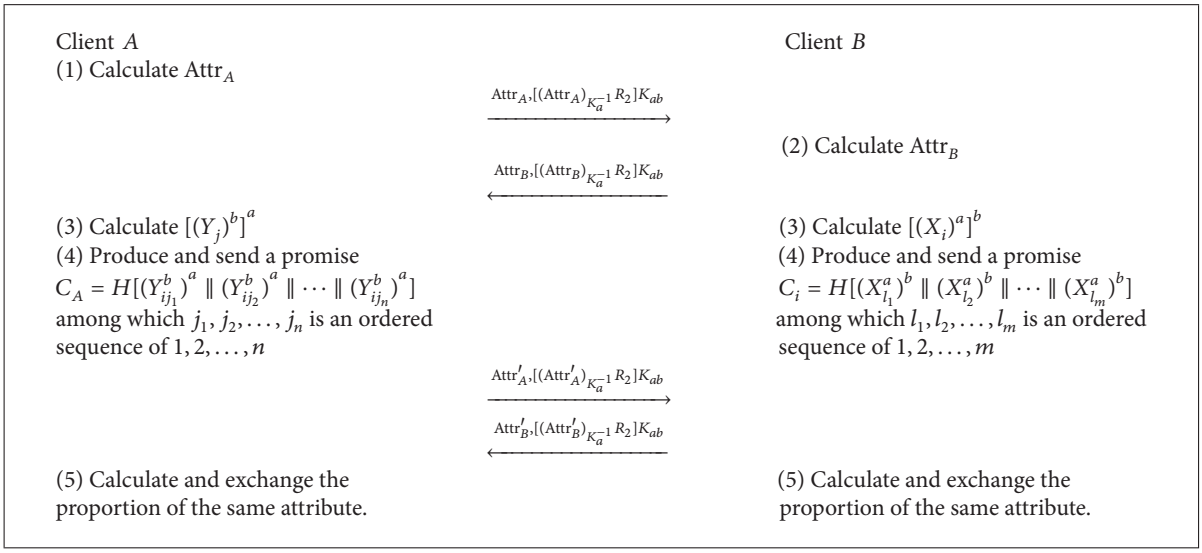
Our mutual authentication protocol is developed on the basis of STS workstation protocol. After adding the identity and using a random number timestamp, the identification is completed. This protocol can also be used to defend man-in-the-middle attack.

Matching Protocol. In order to achieve fine-grained friend recommendation, the similarity of common properties with coarse grained recommendation is calculated. The mutual protocol must be done before matching. The protocol is presented as shown in Box 3.

$\mathrm{Attr}_A = (X_1)^a \parallel (X_2)^a \parallel \cdots \parallel (X_n)^a$, $\mathrm{Attr}_i = (Y_{i1})^b \parallel (Y_{i2})^b \parallel \cdots \parallel (Y_{in})^b$, and i denotes all the users that are recommended to A. $\mathrm{Attr}'_A = (X^a_{l_1})^b \parallel (X^a_{l_2})^b \parallel \cdots \parallel (X^a_{l_m})^b$ and $\mathrm{Attr}'_i = (Y^b_{ij_1})^a \parallel (Y^b_{ij_2})^a \parallel \cdots \parallel (Y^b_{ij_n})^a$.



Box 3: Matching protocol.



Box 4: Attribute exchange protocol.

The core of the matching protocol is the principle of commutative encryption of messages. Pairing of interaction acquires the amount of common properties through two encrypted comparisons of each property.

Meanwhile, the confusing operation is added to ensure the equity and security. In the last step, common properties of both sides are compared; if they do not match, then we turn into arbitration.

Common Property Exchange Protocol. This protocol is used for pairing of users to exchange the detail of common properties. The mutual protocol must be done before exchanging properties. Box 4 shows this protocol.

The core of this protocol is also the commutative encryption, but it does not include a confusing operation. Pairing of interaction can get the detail of properties by the number. Similarly, common properties of both sides are compared in the last step, and hence we decide whether to have arbitration.

5. Security Principles

In this section, we evaluate the security of the proposed scheme under the security protocols we proposed in Section 4. Our protocols are designed based on the Dolev-Yao security model [23].

5.1. Definitions

Definition 1 (ignorable function). Function $y : N \to R$ is an ignorable function, when $\forall$ polynomial p, $\exists n \in N : \forall k > n$,

$$0 \le y(k) < \frac{1}{p(k)}. \tag{6}$$

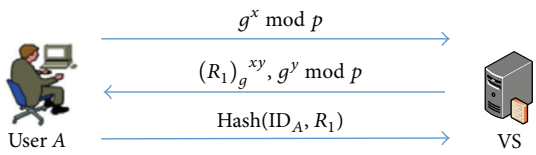


Figure 2: One-way authentication protocol.

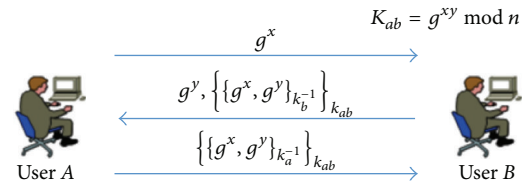


Figure 3: Mutual authentication protocol.

Definition 2 (probabilistic polynomial (PP)). A language L is in PP if and only if there exists a probabilistic Turing machine M, such that

M runs for polynomial time on all inputs;

for all x in L, M outputs 1 with probability strictly greater than 1/2;

for all x not in L, M outputs 1 with probability less than or equal to 1/2.

Definition 3 (computationally indistinguishable). Probability collectives $\{X_n\}$ and $\{Y_n\}$ are computationally indistinguishable if $\exists$ probabilistic polynomial Turing machine D, a large enough integer n, and any polynomial p:

$$\left|\Pr\left[D\left(X_n\right) = 1\right] - \Pr\left[D\left(Y_n\right) = 1\right]\right| < \frac{1}{p\left(n\right)}. \tag{7}$$

For D, $\{X_n\}$ and $\{Y_n\}$ are the same. D cannot get any information of $\{X_n\}$ from $\{Y_n\}$ and vice versa.

Definition 4 (Decisional Diffie-Hellman (DDH) hypothesis). q is a prime number, G_q is a cycle group ordered q, and g is a generator of G_q. And hence, $a, b, c \in_R Z_q$. The distributions of (g, g^a, g^b, g^{ab}) and (g, g^a, g^b, g^c) are computationally indistinguishable.

5.2. Security Analysis. The system security faces a number of threats, such as man-in-the-middle attack, passive wiretapping, property modification, and malicious match by abnormal users. We analyze four threats on our system.

5.2.1. Man-in-the-Middle Attack. The user should register on the VS first. VS matches the properties and sends back the coarse recommendation result. The system implements the one-way authentication protocol to defend against the man-in-the-middle attack. The process is shown in Figure 2.

The fine-grained friend recommendation of our system is based on matching protocol based on commutative encryption function. In order to defend against the man-in-the-middle attack, we implement mutual authentication protocol when the communication channel is established. The process is shown in Figure 3.

One-way authentication protocol achieves the bidirectional key and unidirectional entity confirmation. Mutual authentication protocol achieves the bidirectional key and bidirectional entity confirmation. They complete forward secrecy and nonrepudiation, which can defend against basic man-in-the-middle attacks including replay attack, reflection attack, and interleaving attack.

5.2.2. Passive Wiretapping. Passive wiretapping is when attackers are trying to obtain users information through the communication channel. The register process, when VS is matching the properties and sending back the coarse recommendation result, is threatened by passive wiretapping.

To avoid passive wiretapping, the system uses encrypted properties. Key agreement in the authentication protocols will encrypt the value of the properties, which will efficiently defend against passive wiretapping.

Communication channel's general model shown in Figure 4 shows that the source is discrete and memoryless with entropy H_s. The "main channel" and the "wiretap channel" are discrete memoryless channels with transition probabilities $Q_M(\cdot \mid \cdot)$ and $Q_w(\cdot \mid \cdot)$. The source and the transition probabilities Q_M and Q_w are given and fixed. The encoder, as shown in the figure, is a channel with the K vector S^K as input and the N vector X^N as output. The vector X^N is in turn the input to the main channel. The main channel output and the wiretap channel input are Y^N. The wiretap channel output is Z^N. The decoder associates a K vector $\widehat{S}^K$ with Y^N, and the error probablity $P_e = (1/K)\sum_{k=1}^{K} \Pr\{S^K \neq \widehat{S}^K\}$. The source sends a data sequence $S_1, S_2, \ldots$, which consists of independent copies of the binary random variable S, where $\Pr\{S = 0\} = \Pr\{S = 1\} = 1/2$. The encoder examines the first K source bits $S^K = (S_1, \ldots, S_k)$ and encodes S^K into a binary N vector $X^N = (X_1, \ldots, X_N)$. X^N in turn is transmitted perfectly to the decoder via the noiseless channel and is transformed into a binary data stream $\widehat{S}^K = (\widehat{S}_1, \ldots, \widehat{S}_k)$ for the delivery to the destination. The wiretapper observes the encoded vector X^N, through a binary symmetric channel with crossover probability P_0 $(0 < P_0 \leq 1/2)$. The corresponding output at the wiretap is $Z^N = (Z_1, \ldots, Z_N)$, so that, for $x, z = 0, 1$ $(1 \leq n \leq N$, $\Pr = \{Z^n = z \mid X^n = x\} = (1 - P_0)\delta x,\ z + P_0(1 - \delta x, z))$, with $\Delta \triangleq (1/K)H(S^K \mid Z^N)$, and the transmission rate is KH_S/N source bits per channel input symbol.

As shown in Figure 5, Alice and Bob communicate with each other. The transmission probability is $P_{YZ/X}(y, z/x)$. Supposing that the channel has no memory, the transmission probability of length of sequence n is

$$P\left(Y^n, \frac{Z^n}{X^n}\right) = \prod_{i=1}^{n} P_{YZ/X}\left(y_i, \frac{z_i}{x_i}\right). \tag{8}$$

Alice sends a common message M_0 to Bob and Eve and sends a private message to Bob. The codeword $A(2^{nR0}, 2^{nR1}, n)$ is defined by these:

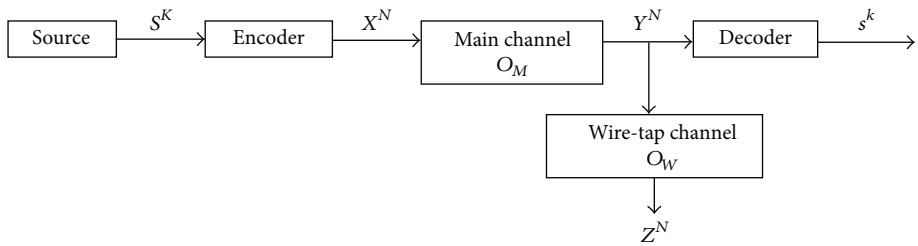


Figure 4: Communication channel general model.

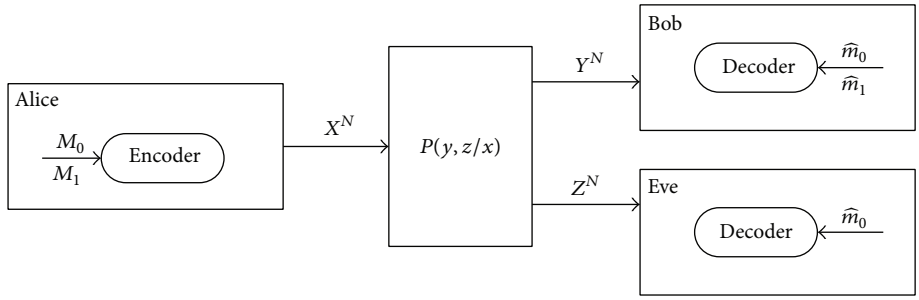


Figure 5: Secret information.

(1) $M_0 = \{1, 2, \ldots, 2^{nR0}\}$ and $M_1 = \{1, 2, \ldots, 2^{nR1}\}$.

(2) An encoding function: $f_n : M_0 \times M_1 \rightarrow \chi^n$, $(m_0, m_1) \in M_0, M_1$, $X^n \in \chi^n$.

(3) Two decoding functions: $g_n : Y^n \rightarrow M_0 \times M_1$ and $h_n : Z^n \rightarrow M_0$. Y^n get $(\widehat{m}_0, \widehat{m}_1)$, Z^n get $\widehat{m}_0$.

Attackers get the signal M_1's with uncertainty: $(1/n)H(M_1/Z^n)$; it needs a codeword $(2^{nR0}, 2^{nR1}, n)$; for any $\varepsilon > 0$, the set of rate (R_0, R_1, R_e) should satisfy

$$P\left[g_n\left(Y^n\right) \neq \left(M_0, M_1\right) \text{ or } h_n\left(Z^n\right) \neq M_0\right] < \varepsilon$$

reliable condition,

$$\frac{1}{n}H\left(\frac{M_1}{Z^n}\right) \geq R_e - \varepsilon \tag{9}$$

Conditions of confidentiality.

5.2.3. Property Modification by Abnormal Users. Some malicious users do not follow the protocols. They send fake message stream to get more details and more private information than the normal users. To defend against this kind of attack, the matching protocol confuses the property information sequence; thus, the information will not be leaked by property modification.

Users' behavior analysis is used to find malicious users. Behavior analysis is a technique that can show whether and how strongly one user is similar to other users. In our method, we are using two types of behavior analysis to find malicious users: (1) behavior analysis of a single user with other products and (2) behavior analysis of multiple user IDs with commonly rated products. To analyze behavior of users A and B, we have used the cosine similarity method. If A and B are the rating values of common user IDs rated for a common product, the cosine similarity, θ, is defined as

$$\text{similarity} = \cos(\theta) = \frac{\vec{V}(A) \cdot \overrightarrow{V}(B)}{\left|\overrightarrow{V}(A)\right| \cdot \left|\overrightarrow{V}(B)\right|}. \tag{10}$$

The resulting similarity ranges from 0, usually indicating independence, to 1, meaning exactly the same, with values in between indicating intermediate similarity or dissimilarity.

Malicious users and attacks have been mostly considered from a system perspective for particular protocols or algorithms. We use a game theoretic model to explain abnormal users. The network is modeled as an undirected graph $G = (V, L)$, where each node in V corresponds to one user. An edge $(i, j) \in L$ means that there is a communication link between the users corresponding to nodes i and j. The set of neighbors of user i, denoted by N_i, is the set of users j such that there exists an edge (i, j) : $N_i = \{j \in V \mid (i, j) \in L\}$. The neighbors of user i are also called adjacent nodes to i. Since the graph is undirected, the neighbor relationship is symmetrical: $j \in N_i \Leftrightarrow i \in N_j$. In order to have a model with asymmetric links, the assumption for an undirected graph can be dropped, but we believe the extension to be straightforward. We denote the set of bad users by V_B and the set of good users by V_G. It holds that $V_B \cap V_G = \emptyset$ and $V_B \cup V_G = V$. We will be using the term type of a user for the property of being good or bad (see Figure 6).

Users have a choice between two actions: C (for cooperate) and D (for defect). When all users choose their actions, each user receives a payoff that depends on three things: his own action, his neighbors' actions, and his own type (but not his neighbors' types). The payoff is decomposed as a sum of payoffs, one for each link. Each term of the sum depends on the user's own action and the action and type of his neighbor

Bad

Good		C	D
	C	$N-E, E-N$	$-E, E$
	D	$0, 0$	$0, 0$

Good

Good		C	D
	C	$N-E, E-N$	$-E, 0$
	D	$0, -E$	$0, 0$

Figure 6: The two games that can take place on a link: good versus bad and good versus good.

along that link. Observe that the user is playing the same action against all neighbors. The payoff of user i is denoted by $R_i(a_i \mid t_i)$, when i's action is a_i and i's type is t_i. We extend and slightly abuse this notation to denote by $R_i(a_i a_j \mid t_i)$ the payoff for i when j is a neighbor of i and j's action is a_j. So, the decomposition of i's payoff can be written as

$$R_i(a_i \mid t_i) = \sum_{j \in N_i} R_i(a_i a_j \mid t_i). \tag{11}$$

Users have incentives and disincentives to cooperate; we model both of them in a game theoretic fashion, with appropriate payoffs (N and E).

5.2.4. Malicious Match by Abnormal Users. During the matching process based on commutative encryption function, the attackers can modify its properties to scan the recommended friend's common browsing behaviors. Because the matching protocol is based on the similarity of the user's properties, that is, browsing behavior, the abnormal users can change their properties to get detailed private information of one recommended friend.

To avoid such attack, the system designed the arbitration protocol to detect the abnormal users.

The arbitration protocol is as follows:

(1) In one-way authentication protocol, users will have key agreement with the VS and get a key g^{xy}. The VS gets $(\text{Attr}_i)^y$ from the key.

(2) Both the semihonest user and the recommended friends send their Attr_i, and the VS calculates $(\text{Attr}_i)^y$ using its private key.

(3) Calculate the hash of $(\text{Attr}_i)^y$ by the key from server and clients. Then, the abnormal user can be found.

Then, we analyze the security of the matching protocol based on commutative encryption function.

Let Alice be the protocol initiator and Bob be the person to be recommended.

Theorem 5 (correctness). *When Alice and Bob have the same properties, the system will recommend Bob to Alice.*

Proof. Let Alice have the property set $X = \{X_1, X_2, \ldots, X_n\}$ which indicates browsing behaviors and secret parameter a. Let Bob have the property set $Y = \{Y_1, Y_2, \ldots, Y_n\}$ and secret parameter b. An arbitrary element $m \in (X \cap Y)$.

Alice and Bob calculate m^a, m^b, respectively, and send them to each other. Then, they calculate m^{ab}, m^{ba}. Because the properties sequence is confused, we can only get $m^{ab} = m^{ba}$; according to the DDH hypothesis, m is not clear.

In the second phase, Alice sends $(m^b, (m^b)^a)$ to Bob and Bob sends $(m^a, (m^a)^b)$ to Alice. If $m^{ab} = m^{ba}$, then we can get m from m^a and m^b. Thus, if Alice and Bob have the same properties, the system believes that Bob and Alice have similar browsing behaviors and recommends Bob to Alice. Thus, we can see that the matching protocol based on commutative encryption function is correct. □

Theorem 6. *Matching protocol based on commutative encryption function can defend against passive attack.*

Proof. Semihonest users want to get sensitive private information of other users, such as private key and properties, by analyzing the information they get from the system.

Assume Alice is a semihonest user. She wants to get Bob's private key and properties. During the process, Alice can get Bob's information of $(\{Y_1^b, Y_2^b, \ldots, Y_n^b\}, (X_1^a, (X_1^a)^b), \ldots, (X_m^a, (X_m^a)^b))$. According to the DDH hypothesis, getting Y_j from Y_j^b is hard. So she cannot get b from $(X_1^a, (X_1^a)^b), \ldots, (X_m^a, (X_m^a)^b)$. For the same reason, assume that Bob is a semihonest user. He cannot get any a and $X_1, \ldots, X_n$ of Alice. Thus, in the first phase, both users can only get the size of the common property set. Then, in the second phase, they can only know the common property set without other information. The matching protocol based on commutative encryption function is efficient in defending against passive attack.

To prove that the matching protocol based on commutative encryption function can defend against active attack, we proposed 3 different situations. We analyze the attack from both the initiator and the recommended friends to prove that the matching protocol we proposed can protect the privacy of the user.

Scenario 1. It can occur in the first phase of the protocol. The initiator counterfeits its properties or sends fake message at the last step, which will get false ratio of the intersection. When someone finds the mistake, the protocol can be terminated immediately, and we can find out who the malicious user is from the arbitration protocol. Even if the malicious user is not clear, the common set will be incorrect in phase two. Then, the VS can still find out the malicious user through the arbitration protocol.

Scenario 2. It can occur in the second phase of the protocol. We assume that Alice is the malicious user and Bob is a normal user. At first, Bob sends Alice $(X_i^a, (X_i^a)^b)$. By receiving $(X_1^a, (X_1^a)^b), \ldots, (X_m^a, (X_m^a)^b)$, Alice gets the intersection S_A. Alice counterfeits $(Y_j^b)^a$ (e.g., replaces some part of $(Y_j^b)^a$ by $[(Y_j^b)^a]$, where $Y_j \in S_A$). Bob gets the intersection S_B. Because S_B is a subset of S_A, Alice can get more information than Bob. This kind of attack cannot be detected. In order to avoid the

Table 1: Abilities of defending against attacks.

Protocols	Passive attack	Scenario 1	Scenario 2	Scenario 3
PRUB	√	√	√	√
Xie and Hengartner [6]	√	×	√	√
Agrawal et al. [7]	√	×	×	×

properties modification from the user, we add promise from Alice to Bob

$$C'_A = H\left[\left(Y_1^b\right)^a \| \left(Y_2^b\right)^a \| \cdots \| \left(Y_n^b\right)^a\right]. \quad (12)$$

Using the promise, we can ensure that Alice sends proper information after receiving Bob's $(X_1^a, (X_1^a)^b), \ldots, (X_m^a, (X_m^a)^b)$. If Bob receives different information from Alice, Bob can terminate the process and report to the VS.

Scenario 3. It can occur in the second phase of the protocol. We assume that Alice is the malicious user and Bob is a normal user. Alice modifies its properties (e.g., using $X_i^a, (Z_i)$ to replace $(X_i^a, (X_i^a)^b)$, where Z_i is a random number). Before Bob sends $X_i^a, (Z_i)$, Alice should send the promise of $X_i^a, (Z_i)$ to Bob. Obviously, Bob will think that the protocol is processed normally and compute the information from Alice. If there is no supervision, Alice can get detailed properties of Bob. However, when they commute property set, Bob can find the difference. Then, he can report to the VS. □

We compare our work with some other protocols, shown in Table 1.

Agrawal et al.'s protocol can only defend against the attack from semihonest users. Xie and Hengartner improved Agrawal et al.'s work; the protocol can defend against some attack from malicious users. PRUB improved these protocols by adding promise information and VS validation, which enhance the security of the protocol.

6. Experiment

In this section, we present the performance evaluation of PRUB. We first show how the user uses the system. After that, we evaluate the recommendation performance using the anonymous data from a Chinese ISP.

6.1. Using Process. The user first registers on the VS and the VS requires some data from the user. Figure 7 shows the register interface of the user.

The users are then asked what kind of information they are willing to share and the threshold on how many kinds of information should be the same for the friend recommendation. If the user chooses not to share one catalog of the browsing information, the system will not utilize the catalog to do recommendation.

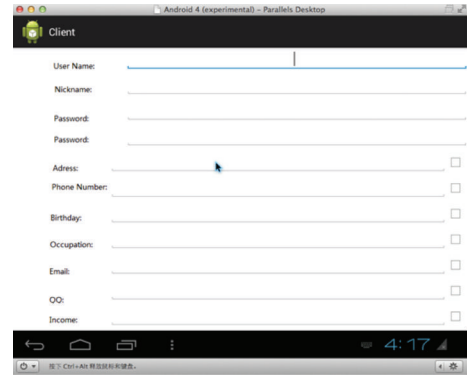


Figure 7: Registration interface.

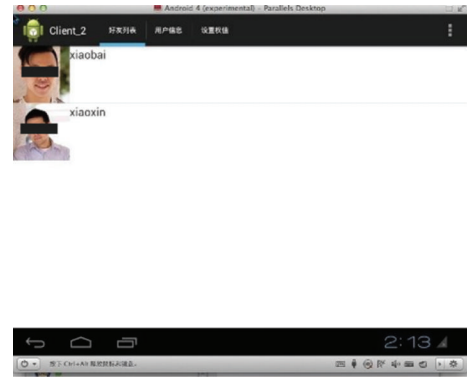


Figure 8: Recommendation list.

After that, the server returns the coarse recommendation result to the client application. Then, the client application scans the recommended friends using the matching protocol we proposed. If a friend shares more common interests, which meet the threshold, the application will add the person to the friend list. Figure 8 shows the friend list.

The user then can choose a friend in the list to exchange the common interests. If the protocol fails, either one can ask for arbitration to find out who the abnormal user is. If it succeeds, the system will give a result with users of the same interest. The result is shown in Figure 9.

6.2. Recommendation Performance. We now evaluate recommendation performance using the anonymous data from a Chinese ISP. The data contained 20000 users' browsing histories for 3 months (October 2013 to December 2013). The users were told that their information would be recorded. Then, the browsing behavior is classified into 12 catalogs. We then computed the ratio of the browsing histories and marked from 1 to 10 for each catalog to indicate the interest of the user. This data is used in the system as properties of the user. A data example of 30 users is shown in Figure 10.

The data is only from the browsing behavior. The friend relationship is not established. According to the Pew Research on Teens, Social Media, and Privacy [24], teen Facebook users have an average (mean number) of 425.4 friends. We assume that every person has about 500 friends and apply k-mean

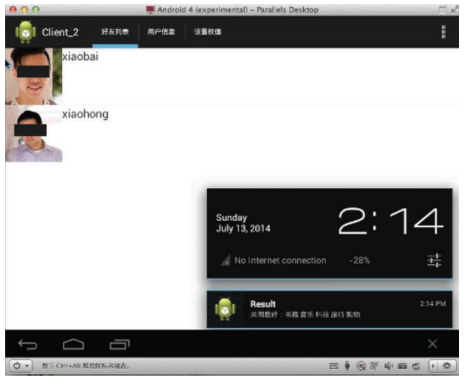


Figure 9: Recommendation result. Users can check what kind of interests they share in common.

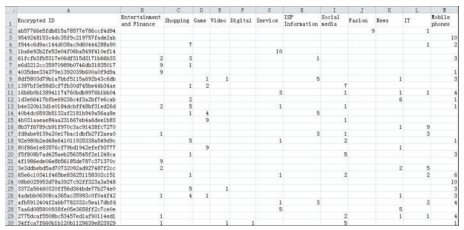


Figure 10: 30-user data from a Chinese ISP.

to form the possible friend relationship of the users from the browsing data we get from a Chinese ISP. Let G_i denote the set of possible friends for each user i.

Then, we apply our recommendation scheme to the ISP data of the user browsing behavior. We then randomly select user to query the system and obtain its friend recommendation results. Let F_i denote the set of recommended friends. The following measurement metrics are used for performance evaluation.

Recommendation Precision. The average of the ratio of the number of recommended friends in the set of possible friends of the query user over the total number of recommended friends is

$$R_p = \frac{\sum_i \left(\text{card}\left(F_i \cap G_i\right)/\text{card}\left(F_i\right)\right)}{500}, \tag{13}$$

where card(·) denotes the number of elements in a set. The dominator is 500 because R_p is the average of 500 users in one experiment.

Recommendation Recall. The average ratio of the number of recommended friends in the set of possible friends of the query user over the number of the sets of possible friends of the query user is

$$R_r = \frac{\sum_i \left(\text{card}\left(F_i \cap G_i\right)/\text{card}\left(G_i\right)\right)}{500}. \tag{14}$$

Using different thresholds, we calculated the average recommendation precision and recommendation recall of the 500 randomly selected users. Figure 11 presents the result.

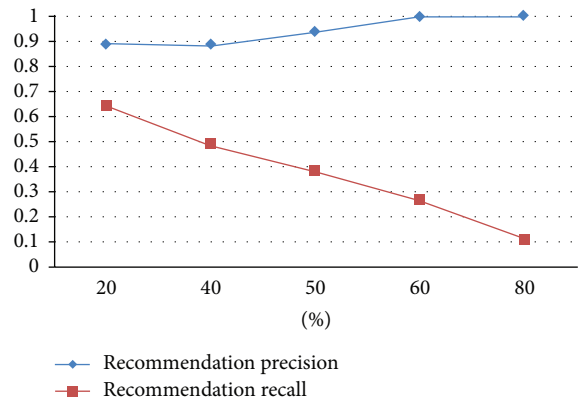


Figure 11: The recommendation precision and recommendation recall of 500 users. The X-coordinate indicates the threshold of matching.

With a higher threshold, the recommendation recall is comparatively low. Friends sharing more alike interests are not easy to find. However, the precision is increasing alongside the threshold. The more the browsing behavior similarities are, the more precise the recommendation result will be.

The experiment shows that our system can achieve relatively high recommendation precision and recommendation recall, and the recommendation system receives remarkable recommendation satisfaction.

7. Conclusion

While enjoying the benefits brought about by the friend recommendation system on the mobile social network, users and researchers begin to notice the personal privacy protection on the social network platforms. In centralized management architectures, all security is ensured by the central server, but, in distributed or hybrid architectures, users can directly exchange information with little or even without the involvement of the server. The security of the network has to be guaranteed by other protocols. In this paper, we presented a secure friend recommendation system PRUB based on user behavior which deploys hybrid management architecture, as a way of reducing the pressure on servers. PRUB can achieve fine-grained recommendation to friends who share the same characteristics without exposing the actual user behavior. In PRUB, the modified matching and authorization protocol can guarantee the privacy. PRUB first uses KNN classification algorithm to do coarse friend recommendation and then uses matching protocol to realize find-grained recommendation. To evaluate the security and performance of PRUB, we theoretically prove that our protocol can defend against attack in the aspect of initiator and matching target, respectively, and we utilize the anonymous data for realistic deployment. The experiment result shows that PRUB not only realizes the fine-grained friend recommendation, but also protects the privacy information of users.

Competing Interests

The authors declare that they have no competing interests regarding the publication of this paper.

Acknowledgments

This work is supported by Fundamental Research Funds for the Central Universities (nos. ZYGX2014J051 and ZYGX2014J066), Science and Postdoctoral Fund in China (2015M572464), and Technology Projects in Sichuan Province (2015JY0178) and the project sponsored by OATF, UESTC.

References

[1] H. Shen, Z. Li, G. Liu, and J. Li, "SOS: a distributed mobile Q&A system based on social networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 4, pp. 1066–1077, 2014.

[2] J. Liu, P. Dolan, and E. R. Pedersen, "Personalized news recommendation based on click behavior," in *Proceedings of the 15th International Conference on Intelligent User Interfaces (IUI '10)*, pp. 31–40, ACM, 2010.

[3] A. C. Squicciarini, F. Paci, and S. Sundareswaran, "Prima: a comprehensive approach to privacy protection in social network sites," *Annals of Telecommunications*, vol. 69, no. 1-2, pp. 21–36, 2014.

[4] T. R. Hoens, M. Blanton, A. Steele, and N. V. Chawla, "Reliable medical recommendation systems with patient privacy," *ACM Transactions on Intelligent Systems and Technology*, vol. 4, no. 4, article 67, 2013.

[5] L. Fang, H. Kim, K. LeFevre, and A. Tami, "A privacy recommendation wizard for users of social networking sites," in *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS '10)*, pp. 630–632, ACM, Chicago, Ill, USA, October 2010.

[6] Q. Xie and U. Hengartner, "Privacy-preserving matchmaking for mobile social networking secure against malicious users," in *Proceedings of the 9th Annual International Conference on Privacy, Security and Trust (PST '11)*, pp. 252–259, Montreal, Canada, July 2011.

[7] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 86–97, June 2003.

[8] M. Moricz, Y. Dosbayev, and M. Berlyant, "PYMK: friend recommendation at myspace," in *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 999–1002, ACM, June 2010.

[9] B. K. Samanthula and W. Jiang, "Structural and message based private friend recommendation," in *Proceedings of the International Conference on Advances in Social Networks Analysis and Mining (ASONAM '12)*, pp. 684–690, IEEE Computer Society, 2012.

[10] L. M. Aiello and G. Ruffo, "LotusNet: tunable privacy for distributed online social network services," *Computer Communications*, vol. 35, no. 1, pp. 75–88, 2012.

[11] M. Li, S. Yu, N. Cao, and W. Lou, "Privacy-preserving distributed profile matching in proximity-based mobile social networks," *IEEE Transactions on Wireless Communications*, vol. 12, no. 5, pp. 2024–2033, 2013.

[12] M. Li, N. Cao, S. Yu, and W. Lou, "FindU: privacy-preserving personal profile matching in mobile social networks," in *Proceedings of the IEEE (INFOCOM '11)*, pp. 2435–2443, IEEE, Shanghai, China, April 2011.

[13] S. Alsaleh, R. Nayak, Y. Xu, and L. Chen, "Improving matching process in social network using implicit and explicit user information," in *Web Technologies and Applications*, pp. 313–320, Springer, Berlin, Germany, 2011.

[14] R. Zhang, Y. Zhang, J. Sun, and G. Yan, "Fine-grained private matching for proximity-based mobile social networking," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '12)*, pp. 1969–1977, IEEE, Orlando, Fla, USA, March 2012.

[15] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology—EUROCRYPT 2004*, pp. 1–19, Springer, Berlin, Germany, 2004.

[16] M. Von Arb, M. Bader, M. Kuhn, and R. Wattenhofer, "Veneta: serverless friend-of-friend detection in mobile social networking," in *Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB '08)*, pp. 184–189, IEEE, Avignon, France, 2008.

[17] L. Shundong, W. Daoshun, D. Yiqi, and L. Ping, "Symmetric cryptographic solution to Yao's millionaires' problem and an evaluation of secure multiparty computations," *Information Sciences*, vol. 178, no. 1, pp. 244–255, 2008.

[18] L. Kissner and D. Song, "Privacy-preserving set operations," in *Advances in Cryptology—CCRYPTO 2005*, pp. 241–257, Springer, Berlin, Germany, 2005.

[19] D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung, "Efficient robust private set intersection," *International Journal of Applied Cryptography*, vol. 2, no. 4, pp. 289–303, 2012.

[20] R. Lu, X. Lin, X. Liang, and X. S. Shen, "Secure handshake with symptoms-matching: the essential to the success of mhealthcare social network," in *Proceedings of the 5th International ICST Conference on Body Area Networks (BodyNets '10)*, pp. 8–15, ACM, Corfu Island, Greece, September 2010.

[21] C. Hazay and Y. Lindell, "Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries," in *Theory of Cryptography*, pp. 155–175, Springer, Berlin, Germany, 2008.

[22] Z. Yang, B. Zhang, J. Dai, A. C. Champion, D. Xuan, and D. Li, "E-SmallTalker: a distributed mobile system for social networking in physical proximity," in *Proceedings of the 30th IEEE International Conference on Distributed Computing Systems (ICDCS '10)*, pp. 468–477, IEEE, Genova, Italy, June 2010.

[23] D. Dolev and A. C. Yao, "On the security of public key protocols," *IEEE Transactions on Information Theory*, vol. 29, no. 2, pp. 198–208, 1983.

[24] M. Madden, A. Lenhart, S. Cortesi et al., "Teens, social media, and privacy, part 2: information sharing, friending, and privacy settings on social media," 2013, http://www.pewinternet.org/2013/05/21/part-2-information-sharing-friending-and-privacy-settings-on-social-media/.