A Privacy Preserving Matchmaking Scheme for Multiple Mobile Social Networks

Yong Wang, Hong-zong Li, Ting-Ting Zhang, and Jie Hou

School of Computer Science and Engineering
University of Electronic Science and Technology of China, 611731
Chengdu, China
cla@uestc.edu.cn

Abstract. Mobile social networks (MSNs) enable users to discover and interact with existing and potential friends in both the cyberspace and in the real world. Although mobile social network applications bring us much convenience, privacy concerns become the key security issue affecting their wide applications. In this paper, we propose a novel hybrid privacy preserving matchmaking scheme, which can help users to find their friends without disclosing their private information from multiple MSNs. Specifically, a user (called initiator) can find his best-matches among the candidates and exchange common attributes with them. However, other candidates only know the size of the common attributes with the initiator. The simulation results indicate that our scheme has a good performance and scalability.

Keywords: privacy preserving, matchmaking protocol, mobile social network, homomorphic encryption.

1 Introduction

With the popularity of personal hand-held mobile devices (e.g., smart phones and PDAs), mobile users can access plenty of Internet services, which brings convenience to users and improves social relationships. Mobile Social Networks (MSNs) provide ad-hoc networking functionality through the Internet, which enables mobile users to search and manage friends, build friendship connectivity, and further disseminate, query and share interesting data sources among them.

Matchmaking can help users to discover and make friends with others who share common attributes (e.g., interests). However, these applications raise a number of privacy concerns [1]. For example, if users' private attributes are directly exchanged with each other, the adversaries may easily collect users' personal information in either active or passive ways, which may be exploited for unauthorized purposes. To protect users' private information, it is essential to make sure that only the minimal personal information is disclosed during the matchmaking process and that the disclosure only goes to as few users as possible.

J. Kołodziej et al. (Eds.): ICA3PP 2013, Part II, LNCS 8286, pp. 233–240, 2013.

In this paper, we propose a novel privacy preserving matchmaking scheme for multiple mobile social networks, which adopts a hybrid architecture to reduce the burden of servers and satisfy certain security requirements. Our matchmaking protocol is based on the polynomial evaluation, which consists of two phases: (1) finding the best matches among numerous users; (2) exchanging the common attributes with them. Homomorphic encryption and (t, w)-Shamir Secret Sharing Scheme [2] are used to guarantee private computation of intersection set. The experimental results indicate the effectiveness of our matchmaking scheme.

2 Related Work

The core component of a matchmaking system is the matchmaking protocol. A matchmaking issue can be described as a private set intersection (PSI) problem or a private cardinality of set intersection (PCSI) problem [1]. A huge body of research have been done on PSI protocols and PCSI protocols, which can be classified into three categories:

In [3], Freedman et al. proposed a PSI protocol which is based on polynomial evaluation for the first time. The homomorphic encryption and balanced hashing are used to guarantee private computation of intersection set. However, the protocol is one way, that is, only the client knows the intersection set while the server knows nothing. So the protocol cannot be used in a distributed environment. Later, Kissner et al. [4] achieved a two-way privacy preserving intersection computation on multisets by employing the mathematic properties of polynomials. Ye et al. [5] extended the scheme proposed in [3] to a distributed private matching scheme by using a secret sharing scheme.

Agrawal et al. [6] proposed a protocol which takes the power function $f(x) = x^e \bmod n$ as communicative encryption and achieves linear complexity. However, it is a one-way protocol and doesn't take the defense to malicious attacks into consideration. Xie et al. [7] revised the protocol to defend against malicious attacks. A two-party PSI protocol in game-theoretic setting using crypto-graphic primitives is built in [8]. Commutative encryption is used as the underlying cryptographic primitive.

Freedman et al. [9] proposed the idea of constructing a set intersection protocol from the oblivious pseudo-random function(OPRF). Revisiting this idea, Hazay et al. [10] utilized specific properties of the Naor-Reingold PRF in order to achieve high efficiency in the presence of both semi-honest and malicious models. Recently, Jarecki et al. [11] presented a very efficient protocol for computing a pseudo random function with a committed key (informally, this means that the same key is used in all invocations), leading to an efficient PSI protocol.

Since most previous protocols are two-way, i.e., both two parties can obtained their intersection set at the end of protocol. Directly applying them to the matchmaking problem may lead to the leak of unnecessary attributes information. We propose a two-phase matchmaking protocol based on polynomial evaluation, which only allows the best-matched users exchange their common attributes with the initiator mutually.

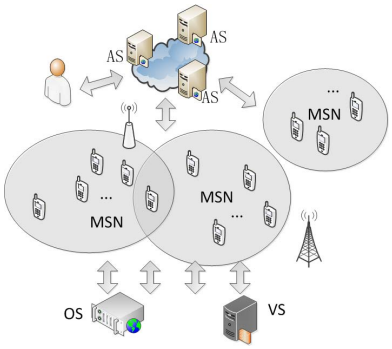

Fig. 1. System architecture

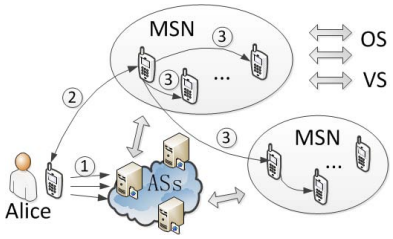

Fig. 2. Procedure of our matchmaking scheme

3 System Design

The system is designed to help a user (called initiator) find his best-matches among multi-parties(called candidates) from multiple MSNs, where the best-match means the user who shares at least ω (a threshold set by the initiator) attributes with the initiator.

3.1 System Architecture

Our matchmaking system consists of four components as shown in Fig. 1.

1) *Mobile users*: Including the initiator and a group of users from different MSNs, each of whom possesses a set of attributes.

2) *The Verification Server* (VS): Which is used to manage users' public keys and attributes information, deal with the deception cases. To initialize a user's identity and attributes, the VS assigns an identity certificate to him and signs his polynomial coefficients created by his attributes. In our matchmaking protocol, system distributes signed identity certificate (ID) to users (e.g., Alice). Each user sends his public key and encrypted polynomial coefficients (we compactly represent them as $\varepsilon_{pk}(P(y)))$ to the VS. Then the VS signs $\varepsilon_{pk}(P(y))$ and returns $sign_{vs}(ID||\varepsilon_{pk}(P(y)))$ to the user.

3) *The Online Server* (OS): Where mobile users can register their public attributes sets and friend lists.

4) *The Anchor Servers* (ASs): Which are semi-honest, having two basic functions: participating in the calculations to reduce the client-side computational burden and detecting malicious attacks.

3.2 Matchmaking Protocol

Fig. 2 shows the procedure of our matchmaking scheme.

Stage 1: Each user distributes his attribute set to w ASs using (t, w)-Shamir Secret Sharing Scheme, where the correctness of each share is publicly verifiable.

Table 1. Computation of $|A \cap B|$

Matchmaking Protocol: Phase 1

Setup: Every user has a public and private key pair (ε, D) for encryption. For Alice, it is (pk_A, sk_A) and for Bob, it is (pk_B, sk_B). Each AS_l also has a public and private key pair (pk_l, sk_l). Each user will do step a)-d). For future reference, we do it for Alice.

a). Alice uses $(t,\ w)$-Secret Sharing scheme to distribute her attributes set $A = \{a_1, a_2, \ldots, a_n\}$ to w ASs verifiably, where the verification vector $\langle \{A_{0,i}\}_{i=0}^{n}, \{A_{1,i}\}_{i=0}^{n}, ..., \{A_{t-1,i}\}_{i=0}^{n} \rangle$ is broadcast in item c) and the share $(\beta_{\ell,0}, \cdots, \beta_{\ell,n})$ sent to AS_l is encrypted using pk_l. Note $(\beta_{\ell,0}, \cdots, \beta_{\ell,n})$ is the vector of coefficients of $F_l(y)$.

b). $\varepsilon_{pk_A}(P(y))$ has been registered at VS who returns a certificate $cert_A = sign_{vs}(Alice||\varepsilon_{pk_A}(P(y)))$ to Alice.

c). Alice authentically broadcasts the following to w ASs:

(I) $cert_A = sign_{vs}(Alice||\varepsilon_{pk_A}(P(y)))$

(II) $\langle \{A_{0,i}\}_{i=0}^{n}, \{A_{1,i}\}_{i=0}^{n}, ..., \{A_{t-1,i}\}_{i=0}^{n} \rangle$

(III) $\{pk_A, \{\varepsilon_{pk_A}(a_1), \varepsilon_{pk_A}({a_1}^2), ..., \varepsilon_{pk_A}({a_1}^\tau)\}, ..., \{\varepsilon_{pk_A}(a_n), \varepsilon_{pk_A}({a_n}^2), ..., \varepsilon_{pk_A}({a_n}^\tau)\}\}$

(where τ is an upper bound of a user's attribute set size) together with a non-interactive zero-knowledge (NIZK)[1] σ that makes sure (II)(III) and $\varepsilon_{pk_A}(P(y))$ in (I) contain the same attribute set A.

d). Alice takes $\gamma_i \overset{R}{\leftarrow} \mathbb{Z}_q^*$ and a random key ζ for a pseudorandom function Φ. She then uses authenticated broadcast encryption [12] to send $(\{\gamma_i\}_1^m, \zeta)$ to w ASs.

Assume Alice is an initiator who wishes to carry out matchmaking with Bob. Let $B = \{b_1, \cdots, b_m\}$ be Bob's attribute set. In the protocol below, symbols defined in Alice's setup are only w.r.t. her (if not exclusively defined, e.g., ζ). For Bob, only c(III) in his setup will be used in the protocol: $\{pk_B, \{\varepsilon_{pk_B}(b_1), \varepsilon_{pk_B}({b_1}^2), ..., \varepsilon_{pk_B}({b_1}^\tau)\}, ..., \{\varepsilon_{pk_B}(b_m), \varepsilon_{pk_B}({b_m}^2), ..., \varepsilon_{pk_B}({b_m}^\tau)\}\}$. So there is no conflict in symbols in the following for Alice and Bob.

Step 1: Alice requests Bob to compute the number of common attributes.

Step 2: Upon receiving Alice's request, Bob asks t selected ASs $AS_{l_1}, ..., AS_{l_t}$ to help.

Step 3: Let $\pi = \Phi_\zeta(Alice, Bob)$ be an encoding of a (pseudo) random permutation on $\{1, \cdots, m\}$. For $j = 1, ..., t$, AS_{l_j} uses Alice's setup and Bob's setup c(III) to do:

(a) for $i = 1, ..., m$, computes $\varepsilon_{pk_B}(F_{l_j}(b_i)) = \varepsilon_{pk_B}(1)^{\beta_{0,l_j}} \cdot \varepsilon_{pk_B}(b_i)^{\beta_{1,l_j}} \cdot ... \cdot \varepsilon_{pk_B}({b_i}^n)^{\beta_{n,l_j}}$.

(b) runs $\pi(\varepsilon_{pk_B}(\gamma_1 F_{l_j}(b_1)), ..., \varepsilon_{pk_B}(\gamma_m F_{l_j}(b_m)))$ to get $(\varepsilon_{pk_B}(\gamma_{\pi(1)} F_{l_j}(b_{\pi(1)})), ..., \varepsilon_{pk_B}(\gamma_{\pi(m)} F_{l_j}(b_{\pi(m)})))$ and sends it to Bob.

Step 4: Let $S = 0$. For $i = 1, ..., m$, Bob:

(a) for $j = 1, ..., t$, computes $u_{\pi(i),j} \leftarrow D_{sk_B}(\varepsilon_{pk_B}(\gamma_{\pi(i)} F_{l_j}(b_{\pi(i)})))$.

(b) computes $g^{\gamma_{\pi(i)} F(0, b_{\pi(i)})} \leftarrow \prod_{j=1}^{t} (u_{\pi(i),j})^{c_j}$, where c_j is appropriate coefficient in Lagrange interpolation for F.

(c) updates $S = S + 1$, if $g^{\gamma_{\pi(i)} F(0, b_{\pi(i)})} = 1$.

Finally, If $S \geq \omega$ (supposedly, $S = A \cap B$), ω is the threshold set by Alice, Bob sends S to Alice.

Stage 2: The initiator (e.g., Alice) broadcasts a matchmaking request to her friends, sets a TTL (Time To Live) on the request packet to determine the hops that the request can be forwarded in the MSNs.

Stage 3: When receiving Alice's request, a receiver will perform the matchmaking protocol (table 1 and 2, starting from step 2 and playing the role of Bob) with Alice. After this, he will randomly forward Alice's request to his friends. This stage will recursively repeat until TTL of the packet decreases to zero.

Table 2. Computation of $A \cap B$

Matchmaking Protocol: Phase 2 (only if $S \geq \omega$)
Step 5: Alice sends $cert_A = sign_{vs}(Alice \| \{\varepsilon_{pk_A}(\nu_0), \varepsilon_{pk_A}(\nu_1), ..., \varepsilon_{pk_A}(\nu_n)\})$ to Bob. $\{\upsilon_j\}_{j=0}^{n}$ is the polynomial coefficients that represent $P(y)$.
Step 6: For $i = 1, ..., m$, Bob computes
(a) $\varepsilon_{pk_A}(P(b_i)) = \varepsilon_{pk_A}(\upsilon_0) \cdot \varepsilon_{pk_A}(\upsilon_1)^{b_i} \cdot ... \cdot \varepsilon_{pk_A}(\upsilon_n)^{{b_i}^n}$.
(b) $\varepsilon_{pk_A}(\lambda_i' P(b_i) + b_i) = \varepsilon_{pk_A}(P(b_i))^{\lambda_i'} \cdot \varepsilon_{pk_A}(b_i)$ and sends $\varepsilon_{pk_A}(\lambda_i' P(b_i) + b_i)$ to Alice, where λ_i' is random in Z_q^*.
Step 7: Let $E = \emptyset$. For $i = 1, ..., m$, Alice computes $\rho_i \leftarrow D_{sk_A}(\varepsilon_{pk_A}(\lambda_i{}' P(b_i) + b_i))$. If $\rho_i \in A$, then adds b_i into E. Finally (supposedly, $E = A \cap B$), if $S = \|E\|$, Alice sends π to Bob, otherwise reject.
Step 8: (a) If Bob received π, he finds $\pi(i)$ such that $g^{\gamma_{\pi(i)} F(0, b_{\pi(i)})} = 1$. Set this collection of $b_{\pi(i)}$ as $A \cap B$.
(b) If Alice refuses to send π, Bob sends the following to $\{AS_{l_j}\}_{j=1}^{t}$:
(i) NIZK proof σ_1 that B used in Step 6 is identical to that encrypted in his setup c(III), where the witness is $\{b_i\}_1^m$, $\{\lambda_i'\}_{i=1}^m$ and randomness of ciphertexts at his setup c(III).
(ii) NIZK proof σ_2 that step 4 (b) is computed correctly, using witness sk_B.
Upon σ_1, σ_2, AS_{l_j} verifies their validity and checks if $S = \sharp$ of i's s.t. $\prod_{j=1}^{t} (u_{\pi(i),j})^{c_j} (= g^{\gamma_{\pi(i)} F(0, b_{\pi(i)})}) = 1$. If all checks pass, then AS_{l_j} sends his share $F_{l_j}(y)$ to Bob. Bob then computes $F(0, y)$ himself and evaluates $F(0, b_i)$ to obtain $A \cap B$.
Note, in step 8(b), if Bob is honest, Alice will leak more information (i.e., $F(0, y)$) than sending π to Bob; if Bob is dishonest, Lemma 4 shows that he can not pass step 8(i)(ii) and hence his disclaim is useless. So we can assume step 8(b) never occurs.

The matchmaking protocol proposed in our scheme contains two phases. Phase 1 is to find the best-matches among numerous candidates. Phase 2 is to exchange the shared attribute set with the best-matches. At the end of phase 1, each candidate can obtain the size of shared attribute set while the initiator knows nothing. Only if a candidate becomes a best match, he will send the shared attribute set size to the initiator. At the end of phase 2, the initiator and each best-match will learn their shared attribute set mutually. Phase 1 of our matchmaking protocol is shown in table 1 and phase 2 is shown in table 2. In both phase 1 and 2, we assume the authentication of each message is guaranteed with the signature from the sender.

3.3 Malicious Detection

Extremely, suppose a participant has only one attribute, then he can learn whether the only attribute is in the initiator's attributes set. To avoid this scan attack, ASs provide a malicious detection mechanism. In phase 1, ASs set a threshold value μ to filter out the users whose attributes are less than μ (e.g., μ can be set as the smallest attributes known by AS). Every user's records at ASs, VS can be updated every moderate long period. This prevents some malicious users to update frequently so as to localize a user's attribute set.

3.4 Computation Cost

The setup contains authenticated broadcast (encryption) and NIZK proof and hence is inefficient. However, this is executed once and will be updated after a long time. It does not affect the efficiency of the matchmaking procedure. Further, NIZK for Bob's disclaim procedure can be assumed to never occur as no one can gain from it. Following this, we can conclude that our matchmaking (steps 1-8) is efficient. If ε is ElGamal encryption [13], then in phase 1, it needs $2m(n+1)$ exps for each AS, $2tm$ exps for Bob; in phase 2, it needs $2(n+1)m$ exps for Bob (note the step 8 only can be obtained by de-permutated the result in step 4 and hence cost is negligible) and m exps for Alice. So we can see that our scheme is reasonably efficient.

4 Simulation and Evaluation

In this section, we evaluate the performance of our matchmaking scheme, the two evaluating metrics are:

1)*Hit rate (Hr)*: *Hr* is calculated as:

$$Hr = \phi/\kappa,$$

where κ is the number of the users in the whole network, whose intersection size with the initiator reach the threshold value ω, ϕ is the actual number of the best-matches found by the initiator. *Hr* indicates the percentage of users can be successfully matched using our scheme.

2)*Message overhead (Mo)*: The messages in our scheme are classified into matchmaking messages and delivery messages. The matchmaking messages are those necessary for carrying out phase 1 and phase 2 protocols. The delivery messages are those responsible for forwarding requests in our scheme. *Mo* is defined as the ratio between the bits number for all delivery messages and that for the whole messages.

We implement our scheme in PeerSim simulator [14]. We select six samples (Sample-1–Sample-6) from Epinions social network datasets which have 574, 977, 1444, 2520, 3613, and 5341 nodes with the average degree 8.52, 11.9, 16.2, 21.1, 23.2, and 26.4 respectively [15]. We choose a prime p of length 1024 bits

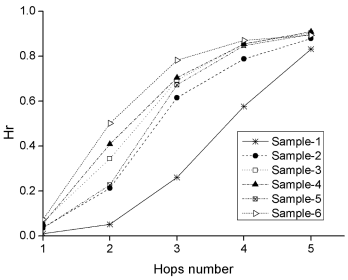

Fig. 3. *Hr* vs. message forwarding hops

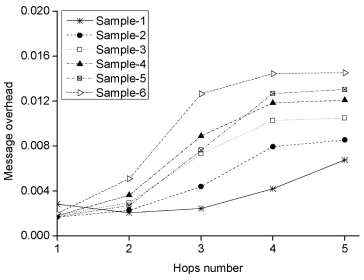

Fig. 4. Message overhead vs. hops number

and use the variant of ElGamal with a modulus length of 1024 bits. Each attribute is represented by 32 bits. We simulate the protocol on 4 PCs, with 1.5 GHz processor and 2G RAM. In order to get more accurate executing time, we repeated each experiments 20 times. The transmission delays are randomly set to be 50-100ms, and the delay of message processing is fixed to be 150ms. Each user has 20 attributes. The threshold value ω is 10. The number of ASs is 3.

The Fig. 3 results show that *Hr* rises to 0.80 averagely when hops number is 4, which indicate that the initiator can find about 80% of his best-matches in the network. Note that Sample-1 has lower *Hr* because of the network scale and connection density, which implies that our scheme may work well under the large-user-based environments. Generally, 3 hops would satisfy users' needs (more than 60% of best-matches can be found except the Sample-1).

The message overhead of our matchmaking scheme is shown in Fig. 4. The message overhead is less than 0.01, i.e., more than 99.9% message contents are used purely for attributes matchmaking. With the hops number increasing, because of the rises of forwarded requested messages, the message overhead also increases. On the other hand, Fig. 4 indicates that the network topological properties affect the message overhead dramatically, the experiments on the six samples show different message overhead variations. The larger and denser the network is, the more overhead messages are needed, which indicate that our scheme may spend more extra energy on large-user-based networks.

The experimental results show the efficiency and scalability of our matchmaking scheme. To find best-matches among a large number of users, the initiator doesn't need to be involved in phase 1. That is, when she sends her request to her friends, she only waits for the responses coming from the best-matches, which can improve user's experience.

5 Conclusion

Matchmaking helps users find their potential friends, but raises serious privacy issues. It is important to develop protocols and schemes to preserving users' privacy in such application scenarios. In this paper, we present a hybrid privacy

preserving matchmaking scheme for MSNs, which can help users to find their potential friends in multiple mobile social networks without leaking their private data beyond necessary.

Acknowledgments. We thank the anonymous reviewers for their helpful comments. This work is supported by a SafeNet Research Award, and by the Joint Funds of the National Natural Science Foundation of China (Grant No.U1230106).

References

1. Li, M., Cao, N., Yu, S., Lou, W.: FindU: Privacy-Preserving Personal Profile Matching in Mobile Social Networks. In: Proc. of Infocom 2011 (2011)
2. Shamir, A.: How to Share a Secret. Communications of the ACM 22(11), 612–613 (1979)
3. Freedman, M.J., Nissim, K., Pinkas, B.: Efficient Private Matching and Set Intersection. In: Cachin, C., Camenisch, J.L. (eds.) EUROCRYPT 2004. LNCS, vol. 3027, pp. 1–19. Springer, Heidelberg (2004)
4. Kissner, L., Song, D.: Privacy-Preserving Set Operations. In: Shoup, V. (ed.) CRYPTO 2005. LNCS, vol. 3621, pp. 241–257. Springer, Heidelberg (2005)
5. Ye, Q., Wang, H., Pieprzyk, J.: Distributed Private Matching and Set Operations. In: Chen, L., Mu, Y., Susilo, W. (eds.) ISPEC 2008. LNCS, vol. 4991, pp. 347–360. Springer, Heidelberg (2008)
6. Agrawal, R., Evfimievski, A., Srikant, R.: Information Sharing Across Private Databases. In: Proc. of SIGMOD, pp. 86–97 (2003)
7. Xie, Q., Hengartner, U.: Privacy-Preserving Matchmaking for Mobile Social Networking Secure Against Malicious Users. In: Proc. 9th Int'l. Conf. on Privacy, Security, and Trust (PST 2011), pp. 252–259 (2011)
8. Rahman, M., Miyaji, A.: Private Two-Party Set Intersection Protocol in Rational Model. Journal of Internet Services and Information Security (JISIS) 2(1/2), 93–104 (2012)
9. Freedman, M., Ishai, Y., Pinkas, B., Reingold, O.: Keyword Search and Oblivious Pseudorandom Functions. In: Kilian, J. (ed.) TCC 2005. LNCS, vol. 3378, pp. 303–324. Springer, Heidelberg (2005)
10. Hazay, C., Lindell, Y.: Efficient Protocols for Set Intersection and Pattern Matching with Security Against Malicious and Covert Adversaries. In: Canetti, R. (ed.) TCC 2008. LNCS, vol. 4948, pp. 155–175. Springer, Heidelberg (2008)
11. Jarecki, S., Liu, X.: Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection. In: Reingold, O. (ed.) TCC 2009. LNCS, vol. 5444, pp. 577–594. Springer, Heidelberg (2009)
12. Fiat, A., Naor, M.: Broadcast Encryption. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 480–491. Springer, Heidelberg (1994)
13. Elgamal, T.: A Public Key Cryptosystem and a Signature Scheme Based on Discrete Logarithms. IEEE Transactions on Information Theory IT-31(4), 469–472 (1985)
14. PeerSim: A Peer-to-Peer Simulator (July 23, 2011), `http://peersim.sourceforge.net/`
15. Stanford Network Analysis Platform (July 23, 2011), `http://snap.stanford.edu/data/soc-Epinions1.html`