Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2014, Article ID 637985, 12 pages
http://dx.doi.org/10.1155/2014/637985

Research Article

An Authenticated Privacy-Preserving Mobile Matchmaking Protocol Based on Social Connections with Friendship Ownership

Shin-Yan Chiou and Chi-Shiu Luo

Department of Electrical Engineering, School of Electrical and Computer Engineering, College of Engineering, Chang Gung University, 259 Wen-Hwa 1st Road, Kwei-Shan, Tao-Yuan 333, Taiwan

Correspondence should be addressed to Shin-Yan Chiou; ansel@mail.cgu.edu.tw

Received 28 May 2013; Revised 29 October 2013; Accepted 30 October 2013; Published 16 February 2014

Academic Editor: Wang Xing-yuan

The increase of mobile device use for social interaction drives the proliferation of online social applications. However, it prompts a series of security and existence problems. Some common problems are the authenticity of social contacts, the privacy of online communication, and the lack of physical interaction. This work presents mobile private matchmaking protocols that allow users to privately and immediately search the targets which match their planning purposes via mobile devices and wireless network. Based on social networks, the relationships of targets can be unlimited or limited to friends or friends of friends. It considers the privacy of users and the authenticity of friendships. The privacy means that no private information, except chosen targets, is leaked and the authenticity that signifies no forgery relationships can be successfully claimed. It applies to many applications such as searching for a person to talk to, to dine with, to play games with, or to see a movie with. The proposed scheme is demonstrated to be secure, effective, and efficient. The implementation of the proposed algorithms on Android system mobile devices allows users to securely find their target via mobile phones.

1. Introduction

Recently, online social networks (OSN) have received a great deal of attention. They provide online communities of users for information sharing. They also change the way people communicate and interact. Facebook, LinkedIn, Myspace, Flicker, Plurk, and Twitter, for instance, are successful applications of social networking services.

However, personal information in OSN is shared among group contacts. Due to the private nature of the shared information, data privacy is an indispensable security requirement in OSN applications. For solving the privacy-related problem, scholars use some valuable methods such as oblivious transfer (OT) [1], identity-based encryption (IBE) [2], searchable encryption [3], privacy-preserving profiles searching (PPPS) [4], access-right revocable scheme [5], middleware for mobile social networking [6], privacy-preserving matchmaking protocol [7], and decentralization-based scheme [8].

Beside privacy consideration, authentication [9] is also an important issue for matchmaking schemes, and authentication protocols, such as password-based authentication schemes [10, 11], are required. However, prior to password authentication, key establishment and key agreement [12–16] are needed as well. The first unauthenticated key agreement protocol based on asymmetric cryptographic techniques was proposed by Diffie and Hellman [17, 18]. Later, some authenticated key agreement [19–24] and anonymous key agreement [25, 26] protocols were developed and proposed.

MobiClique [6], a mobile social networking middleware, let users' smartphones broadcast beacons to nearby devices to show their owners' information. MobiClique users download their profile information from Facebook to their devices and send this information to any Bluetooth device nearby for performing a matching. This approach reveals personal private information to anyone.

Meet Gatsby [27] and Loopt [28] are interesting websites which can find nearby people with shared interests. They require a trusted server that participates in each matchmaking operation. The server knows the interests and current location of each user and performs matchmaking based on this information. This approach allows the server to track users.

However, almost all of the applications are centralized and a trusted server is necessary. This centralized deployment results in some limitations. The users have to connect to the server to use the being controlled data. This brings inconvenience because accessing Internet is not always allowable for all users. All private information of a user is stored in the server, so there is the risk of private information leakage. In addition, each user is only authenticated to the sever, so a user has no capability to verify the information provided by another user. As a result, the issues of centralized deployment lead to inconvenience for mobile usage, leakage of private information, and lack of information authenticity between users.

A decentralization-based scheme [8], for privacy issue, suggests a peer-to-peer architecture solution to avoid centralized control for the existing online centralized architecture. It is based on hop-by-hop trust relationships.

FindU [29] is a privacy-preserving personal profile matching schemes for mobile social networks. An initiating user can find the one, from a group of users, whose profile best matches with his/her. Only necessary and minimal information about the private attributes of the participating users is exchanged to limit the risk of privacy exposure.

Xie and Hengartner [7] proposed another privacy-preserving matchmaking scheme for mobile social networking. They extended AgES [30], which uses commutative encryption, to provide the private matching function. The users' interest items are hashed and then compared for achieving privacy preservation. Therefore, a potentially malicious user learns only the interests that he has in common with a nearby user. Although their protocol does not require a trusted server in matchmaking phase, they need a personal interest signer (in interest signing phase) to sign personal interests in advance. Wang et al. [31] proposed another privacy-preserving matchmaking scheme for mobile social networking to enhance the computational performance of [7]. However in their schemes [7, 31], trusted third parties, identity signer and personal interest signer, are required to issue identity certificates and create interests signatures, and social networking friendships cannot be proved directly.

Chiou and Huang [32] and Chiou et al. [33] propose a social-network-based common-friend discovery application which is noncentralized and provides privacy preservation and information authentication. The application aims to find common friends of two users via their personal devices, such as cell phones or PDA, directly, wirelessly and privately.

In this paper, we propose a mobile private matchmaking scheme based on social connection. The special advantage and the novelty is that the proposed scheme is non-centralized and provides privacy preservation, mutual authentication, friendship relation verification, and friendship ownership certification, which guarantee that the matchmaking target is a friend of friend. Via mobile devices, users can use Wi-Fi Direct [34] and free personal area network (PAN) such as Bluetooth [35, 36] or Infrared Data Association (IrDA) [37] to communicate with each other without Internet access requirement. The application keeps personal information private. After executing the application, the only information that users share is their common friend. Furthermore, it authenticates the exchanged information and avoids forging problems. In addition, we implement a simulation prototype based on our proposed scheme on mobile phones running under the Android operating system.

The rest of this paper is organized as follows. In Section 2, we explain terms related to private matching, data ownership certificate, and replay attack resistance. In Section 3, we review related studies. A technical description and construction details for the proposed protocol are, respectively, presented in Sections 4 and 5. Security, efficiency, and performance analysis for the proposed protocol and property comparison between our scheme and related protocols are given in Section 6. Our implementation is described in Section 7, and we provide conclusions and directions for future work in Section 8.

2. Preliminaries

This section reviews terminology related to private matching, including *Private Matching, Data Ownership Certificates, Asymmetric Exchange, and Replay Attack Resistance*.

2.1. Private Matching. Freedman et al. [38] defined a private matching (PM) scheme as a two-party protocol between a client (chooser) C and a server (sender) S. The inputs of C and S are sets drawn from the same domain. At the conclusion of the protocol, C determines which special inputs are shared by both C and S. Li et al. [39] also define the *security requirement* of PM as follows.

Definition 1 (security requirements of PM). Assuming there are two databases, A and B, one is query $Q \subset A$ and one is *matching protocol* which computes $P = Q \cap B$. The scheme is secure and preserves privacy if it satisfies the following requirements.

(1) *Privacy*. Each party can only know P and its input to the *matching protocol*. Aside from this information, no other information is available to either party.

(2) *Nonspoof*. The items in databases A and B are authorized by their respective owners. This means that the user can only query Q if the owner of the specific query item authorizes the item to the user. In other words, the user cannot generate the query item without authorization from the item owner. In addition, the user is required to present proof of this authorization.

2.2. Data Ownership Certificate (DOC). Li et al. [39] define a Data Ownership Certificate (DOC) as an authorization token, which enables a user to prove his or her legitimate

ownership of particular data. The DOC can attest to the data's ownership, provide verifiable element authorization, and prevent spoofing. If the user does not possess the DOC corresponding to the data in question, he or she cannot make queries for the data and convince another person that he or she is a legitimate owner of this data. The DOC can be used with a variety of matching protocols. Two requirements for the security properties of the DOC [39] are defined as follows.

Definition 2 ((DOC) security requirements). Assume Alice and Bob run a matching protocol to obtain information d. The scheme provides security properties from DOC if it satisfies the following requirements.

(1) *Confidentiality*. If Bob is not an authorized owner of d, Bob should not be able to learn that Alice possesses d by running a matching protocol directly with Alice.

(2) *Authenticity*. If Bob is not an authorized owner of d but Alice is an authorized owner of d, Bob should not be able to pollute Alice's matching result; that is, Bob cannot introduce d into the matching result.

Note that *confidentiality* is difficult to achieve cryptographically since we have to consider both privacy and authenticity. Designs that reveal partial information are not acceptable, and schemes that require precomputation by a third party are not desirable. Therefore, sometimes the goal of DOC is referred to as the *reduced confidentiality requirement*.

2.3. Asymmetric Exchange. Assume that Alice and Bob play a private matching game to exchange their lists A and B and, after the game, learn the answer $A \cap B$. Asymmetric exchange (of a private matching game) means that, for *both parties* to learn the answer $A \cap B$, we must *trust* one party (e.g., Alice) to send a *correct matching result* to the other party (e.g., Bob), where Alice is assumed to be the party to make a *final pass* to send an important result to Bob. In symmetric exchange both parties simultaneously identify their common items through the matching protocol. When the two parties play an asymmetric private matching game, they are assumed to honestly report their friend lists and the corresponding computational results. (The HP [39], AgES [30], and FNS [38] schemes are categorized as asymmetric information exchanges.)

2.4. Replay Attack Resistance. A replay attack is a form of network attack in which a valid data transmission is maliciously or fraudulently repeated or delayed [40]. In private matching protocols, *replay attack resistance* means that an adversary (or the originator) who intercepts or eavesdrops data and retransmits it is unable to effectively obtain private information or to successfully pose as a party running a private matching protocol.

3. Related Work

We present representative private matching protocols and friend discovery schemes in this section. The private matching protocols include Hash Protocol (HP) [39], AgES's commutative encryption protocol [30], FNP's polynomial-based protocol [38], and Data Ownership Certificate (DOC) [39], which can be combined with the private matching protocols to prevent spoofing; friend discovery schemes [32] include friend discovery scheme (FDS) and replay attack resistant friend discovery scheme (RR-FDS).

3.1. Private Matching Protocols. In Hash Protocol (HP) [39], a person who wants to query the common items in the other's database computes hash values of items in his own database and so does the person who is queried. Then they exchange these hash values. By this way, they can find the common items without revealing the information of the unmatched items. On the other hand, Agrawal et al. [30] proposed AgES which uses commutative encryption, instantiating as $\mathrm{Enc}_{k2}(\mathrm{Enc}_{k1}(x)) = \mathrm{Enc}_{k1}(\mathrm{Enc}_{k2}(x))$, to privately match items.

Also, Freedman et al. [38] proposed a polynomial-based private matching scheme. They use the property of homomorphic encryption provided by Paillier cryptosystem [41] to achieve stronger privacy. A variant of their scheme, set cardinality private matching, let A know only the set cardinality of $Q \cap B$, $|Q \cap B|$, but the actual items in this set. It's more applicable than previous schemes. After that, Kissner and Song [42] extend FNP scheme to support more functionality. However, these polynomial-based schemes usually have efficiency problem.

Moreover, HP, AgES, and Freedman et al.'s schemes are categorized to asymmetric exchange of information [39], different from symmetric exchange which both parties know the same information in the matching protocol.

Besides those, Li et al. [39] proposed Data Ownership Certificate (DOC) to ensure nonspoof. DOC provides the authorization of items. If a user does not obtain the item and the corresponding DOC, he cannot make the query and convince other users.

3.2. Friend Discovery Schemes. To find the common friends of two users, Chiou and Huang [32] proposed friend discovery protocols, friend discovery scheme (FDS), and replay attack resistant friend discovery scheme (RR-FDS) based on extending the HP and DOC [39] primitive to ensure privacy preservation and prevent mutual friendship spoofing, where RR-FDS adds the property of resistance to replay attacks. Two algorithms, CredentialExchange and FriendshipMatching, are defined in their protocol.

CredentialExchange(U, V). Users U and V exchange credentials with each other. The credentials include friendship certificates to be used in FriendshipMatching.

FriendshipMatching(U, V). Users U and V discover their common friends in a process which preserves privacy,

Table 1: Notations.

Notation	Meaning
pk_A	Public key of user A
sk_A	Private key of user A
id_A	Identity of user A
sn_A	A fixed sufficient long string chosen by user A
$h(\cdot)$	Cryptographic hash functions
$\mathrm{sig}_k(m)$	Signature of m signed using private key k
$\mathrm{ver}_k(m, s)$	True, if s is a valid signature of message m verified using public key k; else false
sig_B^A	$\mathrm{sig}_{sk_B}(h(\mathrm{id}_B)\|\|\mathrm{pk}_A)$
$dm(A)$	$h(\mathrm{id}_A\|\|sn_A)$
$dm(A, r)$	$h(\mathrm{id}_A\|\|sn_A\|\|r)$
$F(A)$	$\{x\|x$ is the friend of user $A\}$
$F(A \cap B)$	$F(A) \cap F(B) = \{x\|x \in F(A)$ and $x \in F(B)\}$
$F(A \cup B)$	$F(A) \cup F(B) = \{x\|x \in F(A)$ or $x \in F(B)\}$
$F(A - B)$	$F(A) - F(B) = \{x\|x \in F(A)$ and $x \notin F(B)\}$
$DM(A)$	$\{dm(x)\|x \in F(A)\}$
$DM(A, r)$	$\{dm(x, r)\|x \in F(A)\}$
$DM(A \cap B)$	$DM(A) \cap DM(B)$
$DM(A \cap B, r)$	$DM(A, r) \cap DM(B, r)$
$\mathrm{En}_{\mathrm{pk}_A}(M)$	Encrypt M using public key pk_A
$\mathrm{De}_{\mathrm{sk}_A}(C)$	Decrypt C using private key sk_A
$E_K(M)$	Encrypt M using symmetric key K
$D_K(C)$	Decrypt C using symmetric key K
$\mathrm{SIG}^A(B)$	$\{\mathrm{sig}_x^A\|x \in F(B)\}$
$\mathrm{SIG}^A(B \cap C)$	$\{\mathrm{sig}_x^A\|x \in F(B)$ and $x \in F(C)\}$
FCL_A	Friendship certificate list of user A
$MH(T, P)$	True, if $\forall T_i \in T \exists p_j \in P \ni p_j \in T_i$; false, else Where P = sets $\{p_j\}$, T = sets $\{T_i\}$.

achieves mutual authentication, certifies mutual friendship, and prevents mutual friendship spoofing.

Since our scheme is designed based on extending the RR-FDS, we now introduce FriendshipMatching of RR-FDS construction.

FriendshipMatching(*Alice*, *Bob*). As shown in Figure 1, in this algorithm Alice and Bob privately match their common friends through the following steps.

(1) Bob $\rightarrow$ Alice, r_B, where r_B is a random number chosen by Bob.

(2) Alice $\rightarrow$ Bob, $DM(\mathrm{Alice}, r_B)$, r_A, where $DM(\mathrm{Alice}, r_B) = \{dm(x, r_B) \mid x \in F(\mathrm{Alice})\}$, $dm(x, r_B) = h(\mathrm{id}_x \| sn_x \| r_B)$, and r_A is a random number chosen by Alice.

(3) Bob compares $DM(\mathrm{Alice}, r_B)$ with $DM(\mathrm{Bob}, r_B)$ to find the matching items $DM(\mathrm{Alice} \cap \mathrm{Bob}, r_B)$, where $DM(\mathrm{Bob}, r_B) = \{dm(x, r_B) \mid x \in F(\mathrm{Bob})\}$ and $DM(\mathrm{Alice} \cap \mathrm{Bob}, r_B) = DM(\mathrm{Alice}, r_B) \cap DM(\mathrm{Bob}, r_B)$.

(4) Bob $\rightarrow$ Alice, $DM(\mathrm{Alice} \cap \mathrm{Bob})$, $\mathrm{SIG}^{\mathrm{Bob}}(\mathrm{Alice} \cap \mathrm{Bob})$, $\mathrm{pk}_{\mathrm{Bob}}$, $\mathrm{sig}_{sk_{\mathrm{Bob}}}(r_A)$, r_B, where $\mathrm{SIG}^{\mathrm{Bob}}(\mathrm{Alice} \cap \mathrm{Bob}) = \{\mathrm{sig}_{sk_x}(h(\mathrm{id}_x) \parallel \mathrm{pk}_{\mathrm{Bob}}) \mid x \in F(\mathrm{Alice} \cap \mathrm{Bob})\}$, and $\mathrm{sig}_{sk_{\mathrm{Bob}}}(r_A)$ is the signature of r_A signed using Bob's private key sk_{Bob}.

(5) Alice compares $DM(\mathrm{Alice} \cap \mathrm{Bob}, r_A)$ with $DM(\mathrm{Alice}, r_A)$ to get the matching items $DM(\mathrm{Alice} \cap \mathrm{Bob})$.

(6) Alice $\rightarrow$ Bob, $\mathrm{SIG}^{\mathrm{Alice}}(\mathrm{Alice} \cap \mathrm{Bob})$, $\mathrm{pk}_{\mathrm{Alice}}$, $\mathrm{sig}_{sk_{\mathrm{Alice}}}(r_B)$.

(7) Bob verifies $\mathrm{sig}_{sk_{\mathrm{Alice}}}(r_B)$ and verifies the signatures of the hash value for the identity of each common friend, concatenating Alice's public key $\mathrm{pk}_{\mathrm{Alice}}$ in $\mathrm{SIG}^{\mathrm{Alice}}(\mathrm{Alice} \cap \mathrm{Bob})$.

Finally, Alice and Bob recognize their common friends $F(\mathrm{Alice} \cap \mathrm{Bob})$.

4. Notations and Technical Preliminaries

4.1. Notations. Table 1 defines the notations used in our proposed protocol. In Table 1, "$\|$" denotes concatenation and sn_A is a fixed sufficient long secret string chosen by user A,

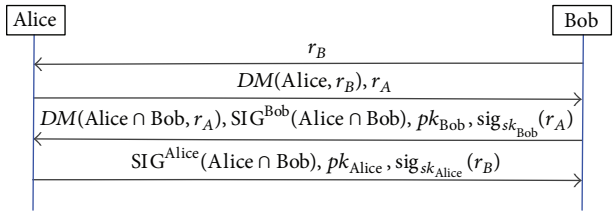


Figure 1: FriendshipMatching(Alice, Bob) of RR-FDS.

where "sufficient long" means the string is long enough to resist brute force or cryptographic attacks. In addition, in this scheme, knowing the secret sn_A implies friendship with user A. Of course, it is not persuasive enough and advanced friendship has to be proven. Moreover, we define $MH(T, P)$ as True if $\forall T_i \in T \exists p_j \in P \ni p_j \in T_i$ and as False if not, where $P = \{p_j\}$, $T = \{\text{sets}T_i\}$. The symbol $T = \{\text{sets}T_i\}$ means target profiles, such as T_1 = "Job: undergraduate," T_2 = "Gender: female," T_3 = "Age: 18–28." The symbol $P = \{p_j\}$ means a personal profiles, such as p_1 = "Job: freshman," p_2 = "Gender: female," p_3 = "Marriage: unmarried," p_4 = "Age: 19," and p_5 = "Interest: watch movie." In this case, $MH(T, P)$ = true.

4.2. Security Requirements. The security requirements of our proposed protocol are as follows.

(1) *Privacy Preservation*. Users can only learn the identity of common friends and nothing else.

(2) *Mutual Authentication*. Users can authenticate one another.

(3) *Mutual Friendship Certification*. Users can prove their friendship to each other. (It is also named *data authenticity.*)

(4) *Mutual Prevention of Friendship Spoofing*. Malicious users are prevented from manipulating these friendship certificates.

(5) *ReplayAttack Resistance*. An adversary (or the originator), who intercepts or eavesdrops the data and retransmits it is prevented from successfully obtaining private information or posing as a party running a private matchmaking protocol.

5. Proposed Scheme

The proposed protocols, mobile private matchmaking (MPM), developed in this paper are based on extending the RR-FDS of Chiou and Huang protocol [32], primitive to ensure privacy preservation, prevent mutual friendship spoofing, and provide friendship discovery. The protocols are defined as three algorithms, Init, CredentialIssue, and AuthObjectMatching. We first describe the syntax of MPM with privacy and authenticity and then present the specification of our protocol.

5.1. Syntax of MPM with Privacy and Authenticity. MPM consists of three algorithms as follows.

Init(1^κ)*Algorithm*. This algorithm is executed once by each user U_i. On input of a security parameter 1^κ, it initializes internal parameters, generates public key pk_{U_i} and private key sk_{U_i}, and clears U_i's friendship certificate list, that is, $\text{FCL}_{U_i} = \phi$ (an empty set).

Credentiallssue(U, V). This is credential issue protocol executed by users U and V. U issues a personal credential Crd_U^V to V, and V issues Crd_V^U to U. Crd_V^U and Crd_U^V are added to friendship certificate lists FCL_U and FCL_V, respectively. The credentials stand for the friendship between U and V. They are used in matching to show friendship and provide the friendship evidence. The inputs of U and V are (id_U, pk_U, sk_U, sn_U) and (id_V, pk_V, sk_V, sn_V), where id_U is the identity of user U, pk_U is the public key of user U, sk_U is the private key of user U, and sn_U is a fixed sufficient long string chosen by user U.

AuthObjectMatching(U, V). This is an authenticated object matching protocol executed by users U and V. In this protocol, U and V hope to find a matched object and their common friendship is also checked. It is designed as two-state mechanism such that U is an *initializer* and V is a *responder*. The inputs of U and V are (T_U, P_U, sn_U, pk_U, sk_U, FCL_U) and (T_V, P_V, sn_V, pk_V, sk_V, FCL_V), where T_U is U's target profile which consists of the profiles of the target object and P_U is U's personal profile which consists of the profiles of user U.

MPM allows users to find a matched object, recognize their common friends, and authenticate to each other. It proves friendship-credential ownership and replay attack resistance. The correctness and security of MPM are defined in Definitions 3 and 4.

Definition 3 (correctness of MPM). Assume that users U and V interact in a MPM protocol with input (T_U, P_U, sn_U, pk_U, sk_U, FCL_U) and (T_V, P_V, sn_V, pk_V, sk_V, FCL_V), respectively, and let π_U and π_V denote the corresponding sessions. By $\text{ID}_\cap$ we denote the set of identities that appears in both FCL_U and FCL_V. MPM scheme is correct if (1) (find out a matched object) π_U and π_V complete in the same state, which is accepted if and only if $MH(T_U, P_V) = MH(T_V, P_U) = true$ or is rejected if U or V gets "not right object" information; (2) (friend discovery) both U and V learn $F(U \cap V)$; (3) (mutual authentication) U and V can authenticate each other; (4) (friendship proof) U and V can prove their friendship of $F(U \cap V)$.

Definition 4 (security of MPM). Assume that users U and V interact in a MPM protocol with inputs (T_U, P_U, sn_U, pk_U, sk_U, FCL_U) and (T_V, P_V, sn_V, pk_V, sk_V, FCL_V), respectively. MPM scheme is secure if (1) (privacy preservation) U and V only learn $F(V \cap U)$ and nothing else (i.e., U learns nothing about $F(V - U)$ and V learns nothing about $F(U - V)$); (2) (mutual authentication) U and V can authenticate to each other; (3) (mutual friendship ownership certification) U and V can prove their friendship to each other via the signatures of

pk_U and pk_V signed using the private keys of their friends; (4) (mutual prevention of friendship spoofing) malicious users are prevented from manipulating these friendship credentials including $\mathrm{SIG}^U(U)$ and $\mathrm{SIG}^V(V)$ even if they obtain FCL_U or FCL_V; and (5) (replay-attack resistance) an adversary, including U and V, who intercepts or eavesdrops the data and retransmits it, fails to obtain private information or to pose as a party U or V running a private matchmaking protocol.

5.2. The Protocol Specification. Based on Definitions 3 and 4, three algorithms Init, CredentialExchange, and AuthObject-Matching of MPM are presented as follows.

Init(1^κ)*Algorithm*. The set-up routine run by each user U mainly consists of the generation of safe RSA [43] parameters. Given security parameter κ, two κ-bit safe primes P and Q are picked randomly. The RSA modulus is set to $N = PQ$, and a pair $e, d \in Z_{\varphi(N)}$ is chosen such that $ed = 1 \pmod{\varphi(N)}$. We denote $\mathrm{pk}_U = e$ is a public key and $sk_U = d$ is a private key. (The underlying cryptosystem can also be ECC [44] or other cryptographic systems).

CredentialExchange ((id_U,pk_U,sk_U, sn_U), (id_V,pk_V,sk_V, and sn_V)) *Protocol*. Users U and V generate personal credentials Crd_U^V and Crd_V^U for each other as follows.

(1) $U \rightarrow V, \mathrm{id}_U, \mathrm{pk}_U$, where id_U is the identity of U and pk_U is the public key of U. The underlying public key cryptosystem can use RSA [43], ECC [44], or other cryptosystems. Note that the key pair pk and sk is generated by each user, not by a trusted third party such as CA.

(2) V computes $\mathrm{sig}_V^U = \mathrm{sig}_{sk_V}(h(\mathrm{id}_V) \parallel \mathrm{pk}_U)$ to prove his friendship ownership to U, where $h(\cdot)$ is a cryptographic hash function, such as SHA1 [45] or MD5 [46], sk_V is the private key of V, and $\mathrm{sig}_{sk_V}(\cdot)$ represents the signature signed using key sk_V.

(3) $V \rightarrow U, \mathrm{id}_V, \mathrm{pk}_V, sn_V, \mathrm{sig}_V^U$, where sn_V is a sufficiently long string chosen by V.

(4) U verifies sig_V^U. If $\mathrm{ver}_{\mathrm{pk}_V}((h(\mathrm{id}_V) \parallel \mathrm{pk}_U), \mathrm{sig}_V^U)$ is true, he computes $\mathrm{sig}_U^V = \mathrm{sig}_{sk_U}(h(\mathrm{id}_U) \parallel \mathrm{pk}_V)$. Else, send "fail" to V and this algorithm fails.

(5) $U \rightarrow V, sn_U, \mathrm{sig}_U^V$.

(6) V verifies sig_U^V. If $\mathrm{ver}_{\mathrm{pk}_U}((h(\mathrm{id}_U) \parallel \mathrm{pk}_V), \mathrm{sig}_U^V)$ is true, finish this algorithm successfully. Else, fail this algorithm.

Finally, U adds $\mathrm{Crd}_V^U = (\mathrm{id}_V, \mathrm{pk}_V, sn_V, \mathrm{sig}_V^U)$ in FCL_U and V adds $\mathrm{Crd}_U^V = (\mathrm{id}_U, \mathrm{pk}_U, sn_U, \mathrm{sig}_U^V)$ in FCL_V, where FCL_U means friendship certificate list of U.

An example of CredentialExchange(Alice, Bob) is shown in Figure 2. Note that sn_U and sn_V should be protected from eavesdropping in CredentialExchange. This algorithm can usually be performed via Bluetooth [35, 36], which provides a basic confidentiality service to thwart eavesdropping attempts on packet payloads exchanged between Bluetooth devices [47]. Otherwise, sn_U and sn_V can be encrypted using public keys, pk_V and pk_U, which can be authenticated via Bluetooth device authentication procedures [47].

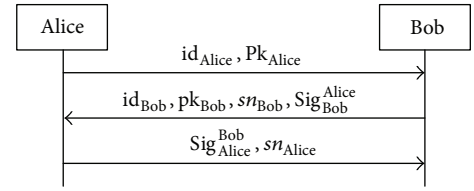


Figure 2: CredentialExchange(Alice, Bob).

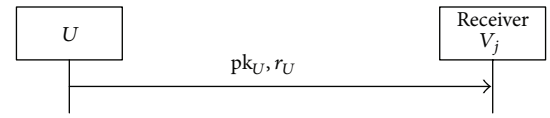


Figure 3: Broadcast of MPM scheme.

AuthObjectMatching ((T_U, P_U, sn_U, pk_U, sk_U, FCL_U), (T_V, P_V, sn_V, pk_V, sk_V, FCL_V)) *Protocol*. Users U and V hope to find a matched object via this protocol. It is designed that U is the *initializer* and V is the *responder*. The protocol is shown as follows. (Also see Figures 3 and 4).

(1) U, who chooses to be an *initializer*, broadcasts pk_U, r_U, where r_U is a random number chosen by U.

(2) V (who chooses to be a *responder*) $\rightarrow U$, $\mathrm{En}_{\mathrm{pk}_U}(r_V)$, $DM(V, r_U \oplus r_V)$, pk_V, $\mathrm{sig}_{sk_V}(r_U)$, where $DM(V, r_U \oplus r_V) = \{dm(V_j, r_U \oplus r_V) \mid V_j \in F(V)\}$.

(3) U gets $r_V = \mathrm{De}_{sk_U}(\mathrm{En}_{\mathrm{pk}_U}(r_V))$ and compares $DM(V, r_U \oplus r_V)$ with $DM(U, r_U \oplus r_V)$ to find the matching items $DM(U \cap V, r_U \oplus r_V)$, where $DM(U \cap V, r_U \oplus r_V) = DM(U, r_U \oplus r_V) \cap DM(V, r_U \oplus r_V)$. If $DM(U \cap V, r_U \oplus r_V) = \emptyset$, U and V have no friends in common. U then sends V a "no match" message and terminates the algorithm. Else, U verifies, if $\mathrm{ver}_{\mathrm{pk}_V}(r_U, \mathrm{sig}_{sk_V}(r_U))$ is true. If it is true, the algorithm proceeds to the next step. Else, U sends a "failure" message and terminates the algorithm.

(4) $U \rightarrow V$, $DM(U \cap V, r_V)$, $\mathrm{SIG}^U(U \cap V)$, $\mathrm{sig}_{sk_U}(r_V)$, $\mathrm{En}_{\mathrm{pk}_V}(K_{UV})$, $E_{K_{UV}}(T_U)$.

(5) V compares $DM(U \cap V, r_V)$ with $DM(V, r_V)$ to get the matching items $DM(U \cap V)$ and then verifies $\mathrm{sig}_{sk_U}(r_V)$ and $\mathrm{SIG}^U(U \cap V)$. If either $\mathrm{ver}_{\mathrm{pk}_U}(r_V, \mathrm{sig}_{sk_U}(r_V))$ or $\mathrm{ver}_{\mathrm{pk}_{V_j}}((h(\mathrm{id}_{V_j}) \parallel \mathrm{pk}_U), \mathrm{sig}_{sk_{V_j}}(h(\mathrm{id}_{V_j}) \parallel \mathrm{pk}_V))$ is false, V sends "failure", and terminates this algorithm, where $V_j \in DM(U \cap V)$. Else, V computes secret key $K_{UV} = \mathrm{De}_{sk_V}(\mathrm{En}_{\mathrm{pk}_V}(K_{UV}))$ and gets $T_U = D_{K_{UV}}(E_{K_{UV}}(T_U))$. If $MH(T_U, P_V) \neq$ true, which means V is not the right object, V sends a "not right object" message and terminates this algorithm, where P_V is V's personal profile. Else, the algorithm proceeds to the next step.

(6) $V \rightarrow U$, $\mathrm{SIG}^V(U \cap V)$, $E_{K_{UV}}(T_V)$.

(7) U verifies $\mathrm{SIG}^{V}(U \cap V)$. If $\mathrm{ver}_{\mathrm{pk}_{U_j}}((h(\mathrm{id}_{U_i}) \parallel \mathrm{pk}_V), \mathrm{sig}_{sk_{U_i}}(h(\mathrm{id}_{U_i}) \parallel \mathrm{pk}_U))$ is true, U computes $T_V = D_{K_{UV}}(E_{K_{UV}}(T_V))$, where $U_i \in DM(U \cap V)$. Else, U sends a "failure" message and terminates the algorithm. If $MH(T_V, P_U)$ is false, which means U is not the right object, U sends a "not right object" message and terminates the algorithm. Else, the algorithm proceeds to the next step.

(8) $U \leftrightarrow V, E_{K_{UV}}(time, location, \ldots)$. U and V find their target objects V and U and negotiate a time, location, and others confidentially.

U and V can recognize their common friends $F(U \cap V)$. In AuthObjectMatching, step 8 can be combined into steps 4 and 6 to reduce transmission times. Notice that no information aside from $F(U \cap V)$ is disclosed. That is, U does not learn any information about $F(V - U)$, and U does not learn any information about $F(V - U)$. Moreover, observe that no trusted centralized server is needed in the proposed protocol.

6. Analysis of Proposed Scheme

6.1. Security Analysis. We analyze the security of our protocols according to the requirements defined in Definition 4.

Privacy Preservation. For U, since each $dm(U_i, TS)$ in $DM(\mathrm{Alice}, TS)$ is the hash values of id_{U_i}, sn_{U_i}, and TS, V or other persons do not know the meaning of $dm(U_i, TS)$ unless he or she has the same pair of id_{U_i} and sn_{U_i}, where $U_i \in F(U)$. Therefore, the information of their noncommon friends is kept private. (Similar to V, the information of his friends is kept private.)

Mutual Authentication. U_A can authenticate U_B from the response messages $\mathrm{SIG}^{U_B}(U_A \cap U_B)$, pk_{U_B} and $\mathrm{sig}_{sk_{U_B}}(r_{U_A})$ since r_{U_A} is a random number chosen from U_A, and pk_{U_B} is signed from $F(U_A \cap U_B)$ in $\mathrm{SIG}^{U_B}(U_A \cap U_B)$.

Mutual Friendship Ownership Certification. From $\mathrm{SIG}^{U_A}(U_A \cap U_B)$, U_A can prove her friendships of $F(U_A \cap U_B)$ to U_B, because $F(U_A \cap U_B)$ signed the pk_{U_A} in $\mathrm{SIG}^{U_A}(U_A \cap U_B)$ and U_B can verify the signature using the public keys of $F(U_A \cap U_B)$.

Mutual Prevention of Friendship Spoofing. Since users U and V have to provide the signatures $(\mathrm{SIG}^{U}(U \cap V), \mathrm{sig}_{sk_V}(r_U))$ and $(\mathrm{SIG}^{U}(U \cap V), \mathrm{sig}_{sk_U}(r_V))$ to each other to prove they are the person who is the friend of $U \cap V$, no one can spoof the friendship without $(U \cap V)$'s signature and his/her own private key.

Replay Attack Resistance. U transmits $DM(U, TS)$ to V using time TS and V transmits $DM(U \cap V, r_U)$ to U using the chosen number r_U. Since the values TS and r_U change, the values $DM(U, TS)$ and $DM(U \cap V, r_U)$ are different in different matching. Therefore, MPM can resist replay attacks.

Table 2: Symbols used in performance analysis.

Symbol	Meaning
a	The number of Alice's friends (e.g., 200)
b	The number of Bob's friends (e.g., 200)
r	The number of common friends (e.g., 2)
l_H	Length of the output of hash function (e.g., 128 bits)
l_S	Length of id, *sn*, pk, and sig (e.g., 1024 bits)
l_{TP}	Length of Target Profile
T_H	The cost of hashing
T_{MH}	The cost of profile matching
T_E	The cost of running symmetric key encryption/decryption algorithm
T_{En}	The cost of running public key encryption/decryption algorithm
T_S	The cost of running public key signature generation algorithm
T_V	The cost of running public key verification algorithm

6.2. Protocol Efficiency and Performance. In this subsection, we analyze the performance of our proposed methods and compare them with other protocols. Table 2 summarizes the symbols used in the comparison. MPM costs are then examined, with communication cost and computational cost, respectively, compared in Table 3. For CredentialExchange, the communication cost is $8l_S$, the computational cost is $2(T_S + T_V + T_H)$, and the total transaction number is 3.

The communication costs of Alice and Bob are $(r+2)l_S + r \cdot l_H + l_{TP}$ and $(r+3)l_S + b \cdot l_H + l_{TP}$, and the total transaction number is 4, where l_{TP} means the length of Target Profile. Here we ignore the costs pk_U, r_U (initial broadcast), and $E_{K_{UV}}(time, location, \ldots)$ (the negotiation between U and V encrypted using the session key K_{UV}).

Both the computational costs of Alice and Bob are approximately $T_S + (r+1)T_V + 2T_{\mathrm{En}}$ since T_S, T_V, and T_{En} cost much time comparing T_H and T_E. Assume each of T_S, T_V, and T_{En} is about 1ms and $r = 2$ then the computational time of Alice and Bob are approximately 6 ms, which is efficient in both computation.

6.3. Property and Performance Comparison. The properties and performances of the proposed protocol MPM are compared with Xie and Hengartner's protocol [7] and Wang et al.'s protocol [31] in Tables 4–6. Comparing Tables 3, 4, and 5, we can see the performance of the proposed scheme is much better in all aspects, where r_I represents the number of common interests or attributes, $l_I/l_M/l_r$ stands for the length of an interest or attribute/a confirm message/the size of intersection set, and *VS* represents the trusted third party Verification Server. All these schemes provide privacy preservation, mutual authentication, and replay attack resistance. However, the schemes of Xie and Hengartner [7] and Wang et al. [31] need thrusted third parties, identity signer, and personal interest signer, to issue identity certificates and create interests signatures. In addition, only our scheme provides the functions of friendship relation and friendship ownership

Table 3: Cost of MPM.

Item	Alice	Bob
Communication cost	$(r+2)l_S + r \cdot l_H + l_{\mathrm{TP}}$	$(r+3)l_S + b \cdot l_H + l_{\mathrm{TP}}$
Computational cost	$T_S + (r+1)T_V + (a+r)T_H + 2(T_E + T_{\mathrm{En}}) + (a+b)\log a$	$T_S + (r+1)T_V + 2bT_H + 2(T_E + T_{\mathrm{En}}) + (b+r)\log b$
Transaction number	2	2

Table 4: Cost of Xie and Hengartner protocol [7].

Item	Alice	Bob
Communication cost	$(2a+2b+r_I+6)l_S + r_I l_I + l_H$	$(2a+2b+r_I+4)l_S + r_I l_I$
Computational cost	$3T_S + (a+b)T_V + T_H + (b+r_I+2)T_E + (a+b)\log a$	$2T_S + (a+b)T_V + (a+r_I+2)T_E + (b+a)\log b$
Transaction number	6	4

Table 5: Cost of Wang et al.'s protocol [31].

Item	Alice	Bob	VS
Communication cost	$(a+3b+5)l_S + l_H + l_r + r_I l_I$	$(3a+b+4)l_S + l_r + r_I l_I$	$2l_M$
Computational cost	$3T_S + bT_V + T_H + (b+r_I)T_E + 2(a+b)\log a + 1$	$2T_S + aT_V + T_H + (a+r_I)T_E + 2(b+a)\log b + 1$	$2r\log r$
Transaction number	5 + 1	4 + 1	2

Table 6: Comparison of properties.

	Xie and Hengartner protocol [7]	Wang et al. protocol [31]	Proposed protocol
Privacy preservation	√	√	√
Mutual authentication	√	√	√
Replay attack resistance	√	√	√
Non-TTP requirement			√
Nonidentity signer			√
Nonpersonal interest signer			√
Friendship relations			√
Friendship ownership certification			√

certification, which guarantee that the matchmaking target is a friend of friend.

7. Implementation

In this work, we implement a simulation prototype based on our proposed schemes, including CredentialExchange (as shown in Figure 2) and AuthObjectMatching (as shown in Figures 3 and 4), on mobile phones running the Android operating system.

To implement our proposed scheme, we use two cell phones for each mobile system. The transmission interface is Wi-Fi. According to our proposed scheme, the prototype has two primary capabilities, credential exchange and mission matching. We assume CredentialExchange is finished via the implementation of Chiou and Huang [32]. In this phase, any two persons can exchange their credential to each other by using their cell phones. We implement AuthObjectMatching of MPM scheme. Anyone can recognize their common friends with other persons by using the credentials exchanged in CredentialExchange and find their matched target with each other by using the profile setup in AuthObjectMatching.

Figure 5 shows the Android mobile phone screens of our implementing prototype. We use JAVA program language to implement them. The types of equipment are Samsung Nexus S, which use Android 2.3 professional operation system. The technological specifications of the equipment are 16 GB ROM, 512 MB RAM, and Cortex-A8 1 GHz CPU, with WiFi network function.

First of all, the splash screen is shown in Figure 5(a), where we can see that it has the function to find a person to have meal with, to play game with, to see a movie with, or to do some other activities with. Clicking the button Meal proceeds to the next screen shown in Figure 5(b), where Server (CF)/Client (CF) denotes a server (initializer)/client (responder) choice with common friend function, and Server (NCF)/Client (NCF) stands for a server (initializer)/client (responder) choice without common friend function.

Assume there are two persons who are going to find a person to have meal with via their mobile devices (as shown in Figure 5(a)). After clicking the button Meal from

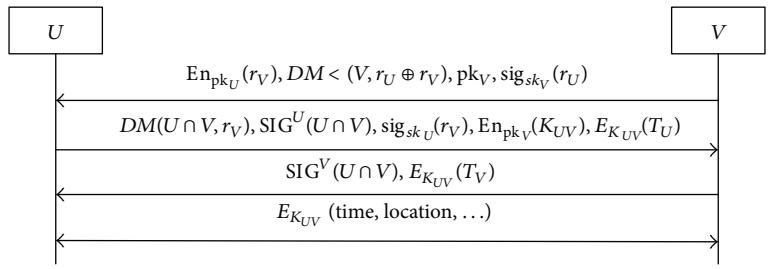
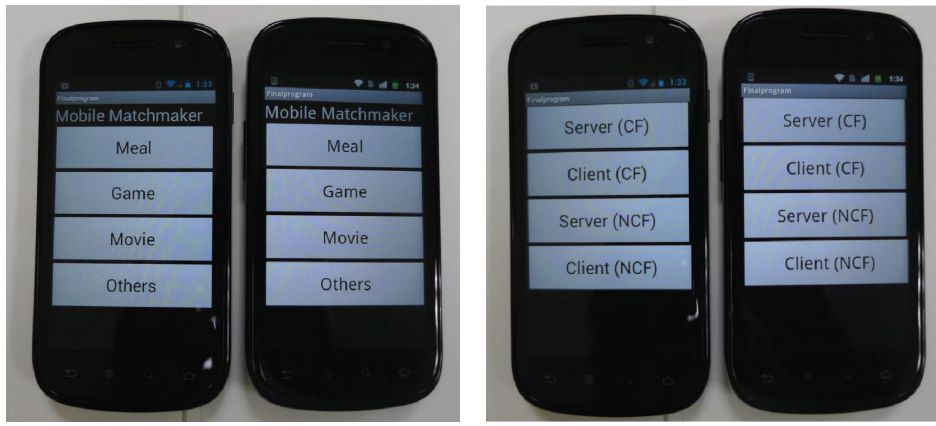


Figure 4: Authenticated object matching.



(a) Splash screen (b) Meal matchmaking

Figure 5: Initial screens of our implementation.

the splash screen (as shown in Figure 5(b)), they can choose to be an *initializer* (or *server*) or *responder* (or *client*) with or without common friend function. Let the left device be a server by clicking Server (CF) button and the right one be a client by clicking Client (CF). Then, the screens of the two devices are shown in Figure 6(a). (If other buttons Server (NCF)/Client (NCF), instead of Server (CF)/Client (CF), are chosen, the scheme directly proceeds to set up personal and target profiles, as shown in Figure 7.)

The two users then click Create Match Data to create their match data dm. After that, client clicks Listen Message to wait for the data-matching message dm_{server} from server and server clicks Send Message to send his message dm_{server} to client. After receiving the message, client clicks Check Common Friends button to check whether they have common friends.

After that, as shown in Figure 6(b), server clicks Listen Message to wait for the response of client, and client clicks Return Message to return its data-matching message dm_{Client} to server. After that, server clicks Check Common Friends button to check whether they have common friends. If they have common friends, as shown in Figures 7(a) and 7(b), they can then set up their Personal Profile and Target Profile to see whether they find each other as their real target to have meal with.

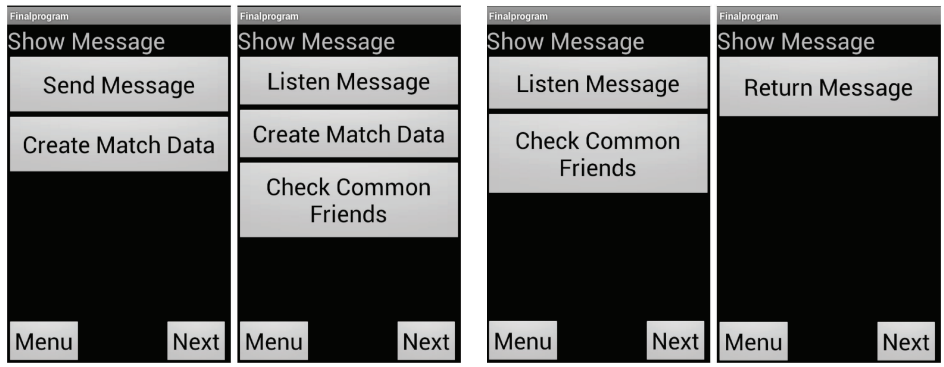
As shown in Figure 7(a), client sets up his or her personal profile and clicks Listen Target Profile to see whether there is a response from server; server sets up his or her target profile and clicks Send Target Profile to send his/her target profile to client. If the personal profile and the target profile matche, server sets up his or her personal profile and clicks Listen Target Profile; client sets up his or her target profile and clicks Send Target Profile (as shown in Figure 7(b)).

If the target profile of client matches the personal profile of server, client then clicks Listen Message to wait for the message from server (as shown in Figure 8.) Server inputs some messages from Enter Message Here box and clicks Send Message to send the input message to client. Similarly, server clicks Listen Message to wait for the message from client. Client inputs some messages from Enter Message Here box and clicks Send Message to send the input message to server.

By using the message exchange, server and client can negotiate for the meal time and place. Moreover, the exchanged message can be encrypted by a session key which can be encrypted by public keys. When the meeting time comes, server and client can authenticate each other via message authentication by the session key or by their public/private keys.

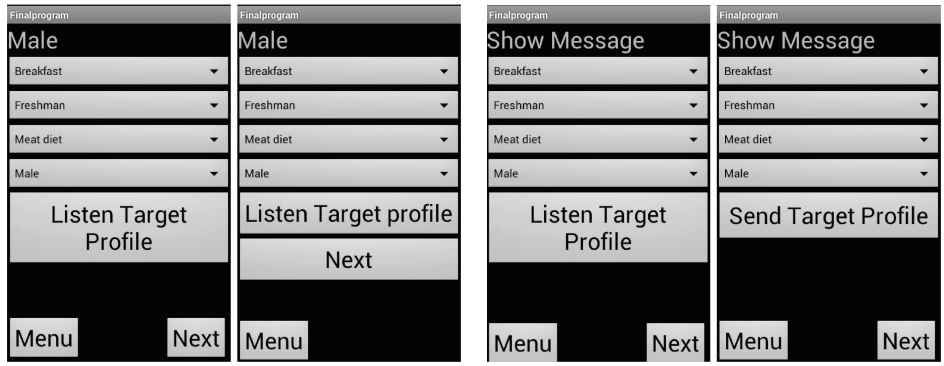
8. Conclusions

In this paper, we present MPM, a mobile matchmaking scheme, which is used not only to find a matched object but also to check whether a common friend exists. In the proposed scheme, privacy preservation, mutual authenticity,



(a) Server sends message (b) Client returns message

Figure 6: Server (a) and client (b) with common friend function.



(a) Server sends target profile (b) Client sends target profile

Figure 7: Setup of personal and target profiles.

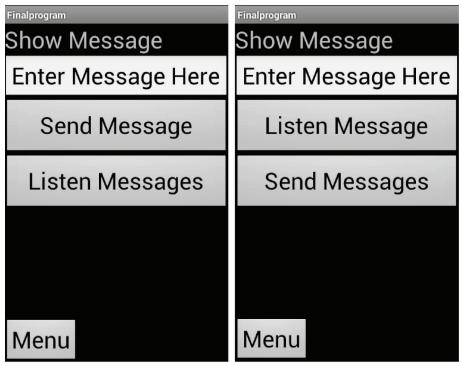


Figure 8: Negotiation.

mutual friendship ownership certification, mutual prevention of friendship spoofing, and replay attack resistance are considered. Comparisons with other approaches show that the proposed schemes provide improved security while performing efficiently in terms of computational and communication costs. The implementation of the proposed algorithms on Android system mobile devices allows users to securely find a matched object.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Science Council under Grant NSC 101-2221-E-182-071. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

References

[1] M. Rabin, "How to exchange secrets by oblivious transfer," Tech. Rep. TR-81, Harvard Aiken Computation Laboratory, 1981.

[2] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology* , pp. 47–53, Springer, Berlin, Germany, 1985.

[3] J. Camenisch, M. Kohlweiss, A. Rial, and C. Sheedy, "Blind and anonymous identity-based encryption and authorised private searches on public key encrypted data," in *Public Key Cryptography—PKC 2009*, pp. 196–214, 2009.

[4] H. Lin, S. S. M. Chow, D. Xing, Y. Fang, and Z. Cao, "Privacy-preserving friend search over online social networks," Cryptology EPrint Archive 2011/445, 2011, http://eprint.iacr.org/.

[5] J. Sun, X. Zhu, and Y. Fang, "A privacy-preserving scheme for online social networks with efficient revocation," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM '10)*, pp. 1–9, March 2010.

[6] A. Pietiläinen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot, "Mobiclique: middleware formobile social networking," in *Proceedings of the 2nd ACM Workshop on Online Socialnetworks*, pp. 49–54, ACM, 2009.

[7] 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, IEEE, July 2011.

[8] L. A. Cutillo, R. Molva, and T. Strufe, "Privacy preserving social networking through decentralization," in *Proceedins of the 6th International Conference on Wireless On-demand Network Systems and Services (WONS '09)*, pp. 145–152, IEEE, February 2009.

[9] X. Wang, D. Zhang, and X. Guo, "Authentication and recovery of images using standard deviation," *Journal of Electronic Imaging*, vol. 22, no. 3, Article ID 033012, 2013.

[10] N. M. G. Al-Saidi, M. R. Md. Said, and W. A. M. Othman, "Password authentication based on fractal coding scheme," *Journal of Applied Mathematics*, vol. 2012, Article ID 340861, 16 pages, 2012.

[11] S. G. Yoo, K. Y. Park, and J. Kim, "A security-performance-balanced user authentication scheme for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 2012, Article ID 382810, 11 pages, 2012.

[12] D. Xiao, X. Liao, and S. Deng, "A novel key agreement protocol based on chaotic maps," *Information Sciences*, vol. 177, no. 4, pp. 1136–1142, 2007.

[13] S. Han, "Security of a key agreement protocol based on chaotic maps," *Chaos, Solitons & Fractals*, vol. 38, no. 3, pp. 764–768, 2008.

[14] E. Chang and S. Han, "Using passphrase to construct key agreement. cbs-is," Tech. Rep., Curtin University of Technology, 2006.

[15] S. Han and E. Chang, "Chaotic map based key agreement with/out clock synchronization," *Chaos, Solitons & Fractals*, vol. 39, no. 3, pp. 1283–1289, 2009.

[16] X.-Y. Wang and J.-F. Zhao, "Cryptanalysis on a parallel keyed hash function based on chaotic neural network," *Neurocomputing*, vol. 73, no. 16–18, pp. 3224–3228, 2010.

[17] W. Diffie and M. E. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[18] C. Wang and J. Luo, "An efficient key-policy attribute-based encryption scheme with constant ciphertext length," *Mathematical Problems in Engineering*, vol. 2013, Article ID 810969, 7 pages, 2013.

[19] X. Wang and J. Zhao, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 12, pp. 4052–4057, 2010.

[20] X.-Y. Wang and Y.-F. Gao, "A switch-modulated method for chaos digital secure communication based on user-defined protocol," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 1, pp. 99–104, 2010.

[21] X. Wang, B. Xu, and C. Luo, "An asynchronous communication system based on the hyperchaotic system of 6th-order cellular neural network," *Optics Communications*, vol. 285, pp. 5041–5045, 2012.

[22] M.-J. Wang, X.-Y. Wang, and B.-N. Pei, "A new digital communication scheme based on chaotic modulation," *Nonlinear Dynamics*, vol. 67, no. 2, pp. 1097–1104, 2012.

[23] H. Liu, X. Wang, and Q. Zhu, "Asynchronous anti-noise hyper chaotic secure communication system based on dynamic delay and state variables switching," *Physics Letters A*, vol. 375, no. 30-31, pp. 2828–2835, 2011.

[24] S. Y. Chiou, "Secure method for biometric-based recognition with integrated cryptographic functions," *BioMed Research International*, vol. 2013, Article ID 623815, 12 pages, 2013.

[25] Y. Niu and X. Wang, "An anonymous key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, vol. 16, no. 4, pp. 1986–1992, 2011.

[26] S.-Y. Chiou, "Authenticated blind issuing of symmetric keys for mobile access control system without trusted parties," *Mathematical Problems in Engineering*, vol. 2013, Article ID 858579, 11 pages, 2013.

[27] Meet Gatsby, "2011 meet gatsby," March 2011, http://meet-gatsby.com/.

[28] Loopt, "loopt," March 2011, http://en.wikipedia.org/wiki/Loopt.

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

[30] 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, ACM, San Diego, Calif, USA, June 2003.

[31] Y. Wang, T. T. Zhang, H. Z. Li, L. P. He, and J. Peng, "Efficient privacy preserving matchmaking for mobile social networking against malicious users," in *Proceedings of the IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom '12)*, pp. 609–615, IEEE, 2012.

[32] S. Y. Chiou and Y. H. Huang, "Mobile common friends discovery with friendship ownership and replay-attack resistance," *Wireless Networks*, vol. 19, no. 8, pp. 1839–1850, 2013.

[33] S.-Y. Chiou, S.-Y. Chang, and H.-M. Sun, "Common friends discovery with privacy and authenticity," in *Proceedings of the 5th International Conference on Information Assurance and Security (IAS '09)*, vol. 1, pp. 337–340, IEEE, September 2009.

[34] Wikipedia, "Wi-Fi Direct," 2012, http://en.wikipedia.org/wiki/Wi-FiDirect/.

[35] Bluetooth specification, "Bluetooth specification," 2012,http://www.bluetooth.com/.

[36] C. Bisdikian, "An overview of the Bluetooth wireless technology," *IEEE Communications Magazine*, vol. 39, no. 12, pp. 86–94, 2001.

[37] Specification, "Infrared Data Association (IrDA) Std," 1998.

[38] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology*, Lecture Notes in Computer Science, pp. 1–19, 2004.

[39] Y. Li, J. D. Tygar, and J. M. Hellerstein, *Computer Security in the 21st Century*, chapter 3, Springer, New York, NY, USA, 2005.

[40] Wikipedia, "Replay attack," October 2012, http://en.wikipedia.org/wiki/Replay_attack/.

[41] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Advances in Cryptology*, vol. 1592 of *Lecture Notes in Computer Science*, pp. 223–238, 1999.

[42] L. Kissner and D. Song, "Privacy-preserving set operations," in *Advances in Cryptology*, vol. 3621 of *Lecture Notes in Computer Science*, pp. 241–257, 2005.

[43] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the Association for Computing Machinery*, vol. 21, no. 2, pp. 120–126, 1978.

[44] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Designs, Codes and Cryptography*, vol. 19, no. 2-3, pp. 173–193, 2000.

[45] N. Standard, "Federal Information Processing Standards Publication 180-1," US Department of Commerce, National Institute of Standards and Technology 131.

[46] R. Rivest, RFC1321: The MD5 Message-Digest Algorithm, RFC Editor United States.

[47] K. Scarfone and J. Padgette, *Guide To bluetooth Security*, NIST Special Publication, 2008.