International Journal of Security and Its Applications
Vol. 9, No. 5 (2015), pp. 217-230
http://dx.doi.org/10.14257/ijsia.2015.9.5.22

Privacy-preserving Attribute Matchmaking in Proximity-based Mobile Social Networks

Solomon Sarpong and Chunxiang Xu

Department of Computer Science,
University of Electronic Science and Technology of China,
Main Building A1-406, No. 2006, Xiyuan Avenue, West Hi-Tech Zone, 611731, Chengdu
sarpong.uestc@gmail.com, chxxu@uestc.edu.cn

Abstract

The major impediments that mostly arise in matchmaking in mobile social networks are ensuring the privacy of users' attributes, finding the intersection of attributes of the matched-pair without revealing any other information, and ensuring that the matched-pair get to know the intersection mutually. Also, in virtually all the existing protocols, the initiator of the matchmaking does not set a threshold number of common attributes an individual should have with him/her before qualifying as a pair. Hence, we propose a hybrid matchmaking cryptographic protocol that will overcome these impediments. In our proposed protocol, an initiator of matchmaking sets a threshold number of common attributes that an individual should have to qualify as a matching-pair. The protocol also ensures that no information about the intersection set is leaked to persons not in the protocol. To further enhance the security and privacy in the protocol, the attributes of the persons our protocol are authorized. The authorization of the attributes is intended to thwart malicious behavior by the persons in the protocol and hence, prevents semi-honest attacks. Furthermore, in this proposed protocol, persons in the protocol get to know the intersection of their attributes mutually.

Keywords: *hybrid, matchmaking, cryptographic, impediments, nonspoofability*

1. Introduction

The advent of mobile phones has brought about a lot of changes in the way we communicate and relate to each other. It has also come along with the readily availability of information. As at now, information can be assessed readily anytime-anywhere. The increasing dependence on anytime-anywhere availability of information and the commensurate increasing fear of losing privacy motivate the need for privacy-preserving techniques in mobile social networks.

Since the inception of social network concept, there have been a lot of improvements. On social networks, a lot of personal and private information are shared. Some of this personal and private information shared are not intended or meant to be available to everyone. As a result persons who should not be privy to this information happen to view and use them. It must be noted that even social networks that are completely open, there is a-disconnect between users willingness to share information and their reaction to unintended persons viewing or using this information [13]. The success of social networks has come along with some of its core and common uses; private matchmaking and finding the characteristics or attributes that are common to two or more persons.

Private matchmaking is interesting as it has conflicting requirements for anonymity and authentication. A private matchmaking protocol allows two or more mutually suspicious parties with matching credentials to locate and authenticate each other without

ISSN: 1738-9984 IJSIA

revealing their credentials or identities to anyone including the matchmaker. Private matchmaking is more than mutual authentication of suspicious parties in that it has further requirements on privacy and efficiently locating. The goals and requirements of a private matchmaking protocol can be motivated by considering a job-referral service. Imagine that a company wants to hire a new vice-president from among one of the current vice-presidents of its competitors. The company does not want to announce the job opening and a person currently employed by a competitor does not want to announce his/her willingness to leave. Neither party wants to reveal information about their wishes unless they know that the other party has a matching wish [5].

Also, there are instances when two parties would like to identify their common customers for a joint marketing exercise, without divulging any additional customers. In this scenario, they would like to ensure that (a) neither party learns more than their own data and must obtain the intersection (if one exists), while neither should learn anything about other set and (b) if one party learns the results of the match, both parties should learn it. Hence, the research community has foreseen the need for mechanisms to enable limited (privacy-preserving) sharing of sensitive information and a number of effective solutions have been proposed. Among them, Private Set Intersection (PSI) techniques are particularly appealing for scenarios where two parties wish to compute an intersection of their respective sets of items without revealing to each other any other information [1].

In their paper, Sang and Shen [8] addressed the issue of privacy preserving set intersection (PPSI) problem, in which each of the N parties learns no elements other than the intersection of their N private datasets. They also proposed an efficient protocol in the malicious model, where the adversary may control arbitrary number of parties and execute the protocol for its own benefit.

Consider two organizations that wish to privately match their data. They want to find common data elements (or perform a join) over two databases without revealing private information. This was the premise of the papers in [3, 15, 20]. The main limitation in these protocols is that, it is asymmetric. Thus if we want both parties to learn the answer, we must trust Alice to send $A \cap B$ to Bob. This asymmetry may be acceptable or even desirable in some scenarios, but may be undesirable in others. In the likely event of two dishonest persons in the protocol, the one receiving the intersection may not truly report the intersection or may even terminate the protocol after knowing the intersection. These scenarios lead to information asymmetry. As a result, the research community has proposed symmetric matchmaking protocols. In these protocols, the parties involved in the protocol obtain the intersection mutually [10, 12, 23, 36].

Most often than not, the sharing of information involves two persons seeking to know if their private sets have any common information. Hence, two main challenges are encountered; (1) how to enable this type of sharing such that the parties learn no (or minimal) information beyond what they are entitled to and (2) how to do so efficiently in real world practical terms [2]. In both asymmetric and symmetric protocols, since each party is not willing to disclose the content of their list, ordinary private set intersection will not be appropriate to use in finding the intersection. In light of this, authorized private set intersection is more appropriate. The main contribution of this research is to formulate a novel hybrid proximity-based matchmaking protocol that is privacy-preserving, efficient and secure against malicious attacks. In our proposed protocol, not only does the initiator find a matching-pair, but the pair that has at least the preset threshold number of common attributes. Furthermore, apart from the matched-pair that is privy to the number and type of their common attributes, no one else does.

The rest of this paper is organized as follows: we take a brief look at private set intersection. In section II, we present related work. Our protocol and the algorithms for the matchmaking are presented in section III. In section IV, we take a look at the security of our algorithm. Finally, we conclude this paper in section V.

1.1 (AUTHORIZED) PRIVATE SET INTERSECTION, (A)PIS

Generally speaking, Private Set Intersection (PSI) is a cryptographic protocol that involves two players, say Alice and Bob, each with a private set. Their goal is to compute the intersection of their respective sets, such that minimal information is revealed in the process. In other words, Alice and Bob should learn the elements (if any) common to both sets and nothing else. This can be a mutual process where, ideally, neither party has any advantage over the other.

In PSI, Alice and Bob having S_A and S_B would wish to find the intersection of their sets. Their wish is to jointly compute the intersection in such a way that reveals nothing of S_A to Bob and S_B to Alice. In other words, both Alice and Bob should learn only the intersection, $S_A \cap S_B$ but nothing more. While this task could be completed with general secure multiparty techniques, it is far more efficient to have a dedicated protocol. This is because no secure multi-party protocol can prevent a party from cheating by changing its input before the protocol starts. A problem common to all these protocols is that the inputs S_A and S_B can be chosen arbitrarily by Alice and Bob [1]. Hence, a dishonest party can therefore insert fabricated elements in its set that s/he suspects the other person might have. The intersection will reveal if the other person indeed has those attributes in his/her set. To address this issue, Authorized Private Set Intersection (APSI) and its variants [4, 16, 32] ensure that each person can only use attributes certified by a certification authority in the intersection protocol.

In particular, we consider the scenario where two persons each hold a set of attributes and wish to find the intersection of their attributes without revealing other attributes that are not in the intersection. In such applications, it is important to ensure that each data item being exchanged is properly authenticated or authorized by a certification authority in the intersection protocol [4]. When authorization is done, it thwarts dishonest behavior. Unless some form of authentication is required, a malicious party can claim possession of fictitious data items, in an attempt to find out whether the other party possesses those data items.

The problem of authentication of mutually suspicious parties is one that is becoming more and more important with the proliferation of distributed systems. A user in a distributed system may not only need to verify the identity of the system, but may require that the system, or another user or node in the system, verifies itself to him/her. Moreover, both sides may require some degree of authentication before they release any information about themselves [7]. The usual solution to such problems (of wanting to know what you have in common without disclosing any other personal secrets) is to consult a trusted third party. However, this may not be practical in a highly distributed situation. For one thing, there may not be a central authority that all parties are willing to trust. On the other hand, such a central authority may exist, but may not be available to all users at all times. One solution to these problems lies in the development of cryptographic matchmaking protocols. That is, protocols in which users with various secrets can verify whether or not their secrets agree without revealing the secrets to each other or a third party.

The goal of certifying the private sets of participants is to restrict their inputs. This reduces the strength of a malicious participant. Bob though malicious will follow the protocol, but as he wishes to learn as much as possible about the private set of Alice, he can strategically populate his private set with all of his best guesses for Alice's private set and hence, his private set will be as large as possible. This maximizes the amount of information Bob learns about Alice's private set. In the extreme case, Bob may claim his private set contains all possible elements, which will always reveal the private set of Alice. He may also vary his set over multiple runs of the protocol, in order to learn more information over time. These attacks are even more powerful when the protocol can be executed anonymously. It must be noted that all these behaviors are permitted in any protocol which allows the participants to choose their inputs arbitrarily [19].

2. Related Work

In this section we will take a look at some previous works on matchmaking for mobile social networking, and then focus on reviewing some cryptographic protocols for matchmaking.

2.1 Mobile Social Networking Applications

In private set intersection problem, two parties each have a set of elements. They compute the intersection of their sets such that no party learns any information other than the intersecting elements. Matchmaking protocol is a private set intersection problem where a matching-pair is made by computing the intersection of their individual attributes.

In matchmaking, most often than not, the introduction of a trusted central server is one of the techniques for addressing such issue. With this, the trusted central server is involved in each step of the matchmaking process. The trusted central server collects personal attributes and location information, computes the intersection and notifies the matched-pair. Such protocol applications can be found in [11, 21, 22]. In the applications such as [28, 29], the websites 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. The use of a trusted central server has got some challenges. That is, it is unlikely that all the users are willing to send their personal information to the server. Also, the centralized server is generally based on connection to the internet, in some application scenarios, users would like to perform matchmaking through multiple communication channels (e.g. Bluetooth). Furthermore, one- point-failure and bottle-neck problems limit the systems' scalability. Also, mechanisms have to be put in place to provide security protection for the centralized server. On the other hand, as the number of users increases, the centralized server may become overloaded as a result, the quality of service may drop.

Another way is the fully distributed technique, which requires no trusted server in the whole matchmaking process, [9, 23, 35]. The operations, such as the distribution of personal attributes data, the computation of the intersection set, and the dissemination of results are performed among multi-parties, without any trusted third party. The attributes of the initiator and the candidates are shared among multi-parties using Shamir Secret Sharing Scheme, the computing of common attributes set are conducted among multi-parties as well [21]. In an application like MobiClique [6], users' smartphones broadcast beacons to nearby devices by using Bluetooth to show their owners information. This approach reveals personal private information to anyone within reach of the Bluetooth.

The third technique in use is a hybrid, where a trusted centralized server is needed only for the purpose of management and verification, and it does not participate in the matchmaking operations. This mechanism can provide efficient matchmaking services with relatively high scalability. In [12, 17, 19, 26, 36], are the protocols based on hybrid mechanisms designed to support privacy preserving attributes matchmaking functions for MSN.

In [15], with two persons involved in matchmaking, at the end of the protocol only one of them, say Alice, computes the intersection set. Thus, this protocol may lead to information asymmetry. Alice may decide not to report the intersection set truly to the other user. Also, Alice may decide not to continue with the matchmaking after knowing the attributes of the other user. Furthermore as the proposed protocol is one way, several malicious attacks can be launched by the parties in the protocols. This protocol was improved upon in [12] by removing the likelihood of malicious attack by persons involved in the protocol. Also, both persons in the protocol perform the intersection set.

These fore-mentioned proposed protocols do not take into consideration if the would-be pair has enough common attributes to qualify to be paired. In [17], there was an improvement in the matchmaking protocols by letting a user (called initiator) find the best match among multi-parties (called candidates). In this protocol, the best match means the user (among other candidates) that has the maximum intersection set size with the initiator. It can be noted that this protocol is not very adequate as the best match does not necessarily mean the pair has got enough common attributes to make a good pair. Privacy-preserving scalar computation of $\vec{a}\square\vec{b}$ was used in [36] to find the number of common attributes two persons have before they are match-paired. In this protocol, the TA has *n* number of attributes and each person in the protocol forms a binary vector of his/her attributes. Let, $\vec{a} = \{\mathrm{a}_1, a_2, \dots, a_n\}$ and $\vec{b} = \{\mathrm{b}_1, b_2, \dots, b_n\}$ represent the private attributes of Alice and Bob respectively. After calculating the scalar product, if $\vec{a}\square\vec{b}$ is greater or equal to a preset number of attributes, Bob becomes the matching-pair of Alice, the initiator.

2.2 Private Matchmaking Protocols

In order to preserve privacy related problems of personal attributes in social networks, some researchers use protocols such as oblivious transfer (OT) [18, 24, 30], identity-based encryption (IBE) [31], searchable encryption [32], privacy-preserving profiles searching (PPPS) [33], access-right revocable scheme [34], middleware for mobile social networking [6], privacy-preserving matchmaking protocol [12], and decentralization-based scheme [23].

Oblivious transfer (OT) is a protocol which allows a server to transfer one of two items to client, such that the client can choose which item s/he wants, keep his/her choice hidden from server, and learn nothing about the other item. OT can be used to construct private set intersection protocols [18, 24, 30]. These were improved upon by Kissner and Song [2]. Using threshold cryptosystem, they proposed efficient protocols to solve privacy preserving set intersection and privacy-preserving set matching problems. The former arises when each member in the protocol wants to learn the intersection of the private dataset. In the later, each party in the protocol wants to learn whether its elements can be matched in any private set of the other parties. In this proposed protocol, efficient computation of intersection, union, and element reduction of multiset operation is achieved by using polynomial representations and employing the mathematical properties of polynomials. Thus, private set intersection (PSI) protocols enable two parties each holding a set of input to jointly compute the intersection of their inputs without leaking any other information. However these are less efficient than specialized protocols, and efficiency decreases when elements are chosen from larger domains hence, cannot be used in a distributed environment. Variants of this protocol can be found in [1, 27]. In [7], there was the use of Shamir Secret Sharing to guarantee the privacy of the intersection set and prevent malicious attacks.

Freedman, *et al.,* [20] introduced the oblivious pseudorandom function based protocols, which implemented private set intersection and private cardinality of set intersection protocol based on pseudorandom function. Hazay and Lindell [24] used efficient secure protocols for set intersection and pattern matching, securely computed the set intersection functionality based on secure pseudorandom function evaluations, in contrast to previous protocols that are based on polynomials. In addition, utilizing specific properties of the Naor-Reingold pseudorandom function, a secure pseudorandom function evaluation in order to achieve secure pattern matching was achieved. In [25], there is an improvement in oblivious pseudorandom function.

In [12], when individuals want to query the common items in their database, they compute the hash values of their individual items. They then exchange the hash values of their individual items. In this way, they are able to find the common items in their intersection without revealing any other items that are not in the intersection. Other researchers use commutative encryption to achieve private set intersection and private cardinality of set intersection. Commutative encryption has the property that $E_{k_1}\left[E_{k_2}\left(x\right)\right]=E_{k_2}\left[E_{k_1}\left(x\right)\right]$. Thus, for two private keys k_1 and k_2, the same encryption will be achieved despite the order of encryption. The main idea of using commutative encryption as a keyed one-way hash function is to generate a mapping for each element x, such that no member of the protocol knows the key. From the above commutative encryption property, either users cannot learn the other party's information outside the intersection because of the lack of the necessary key information. In Agrawal, *et al.,* [15], they suggested the power function, $f_e\left(x\right)=x^e \bmod p$ as an example of a commutative encryption function.

3. Our Protocol

Our proposed protocol will help match-pair seekers find the most appropriate pair in a mutual matchmaking protocol by first allowing them compute the number of attributes they have in common. If the number of common attributes is at least the minimum threshold set by the initiator, they then become a matching-pair. They then exchange their actual attributes they have in common. In order to ensure privacy in this proposed protocol, each person can only know the intersection (if it exits) and his/her input to the matching protocol. Aside from this information, no other information is available to either person. Nonspoofability of the other users' attributes is another characteristic of our protocol. In this protocol, some privacy levels needs to be considered;

Privacy level 1: When algorithm 1 ends the initiator, Alice and each of the other persons in the protocol mutually learn the size of their intersection set. An adversary should learn nothing.

Privacy level 2: At the end of algorithm 2, Alice and the matched-pair(s) in the protocol mutually learn the intersection set (their common attributes) between them. An adversary should learn nothing.

In our protocol, we do not consider the following threats and make the following assumptions:

1) Users keep their private keys safe, so that malicious users cannot steal their private keys to impersonate them or compute their personal attributes;

2) The CA cannot be compromised by attackers;

3) Users trust the person with whom our protocol finds and that the matched-pair will not disclose his/her attribute to others.

The initiator, Alice sets a threshold number of common attributes, $A_{Threshold}$ that an individual(s) should possess to qualify as a match-pair. This is because Alice wants a matching-pair(s) that has at least a certain number of common attributes. In our protocol, in order to prevent malicious persons from manipulating the input set, their private inputs are certified by a certification authority, CA.

3.1 Initial Phase

Our system consists of T users (persons) denoted as $P_1,\ldots,P_T$, each possessing a portable device. Each device of a person in the protocol communicates through wireless interfaces such as Bluetooth or WIFI. Let us assume every participating device is in the communication range with each other. Alice launches the matching process to find a person(s) that has at least the preset threshold number of common attributes, $A_{Threshold}$, among the other persons. Alice has a set of attributes $a_i, i=1,\ldots,n$ whilst each of the $j=T-1$ persons' profile, P_i consists of a set of attributes $b_{jk}, j=1,\ldots,m; k=1,\ldots, p.$ Note that, we assume that the system adopts some standard way to describe every attribute, so that two attributes are exactly the same if they are the same semantically.

As depicted in algorithm 1, the CA generates an RSA key pair, (e,d) and $N=pq$, where $p=2p^{'}+1$ and $q=2q^{'}+1$; $p, q, p^{'}$ and $q^{'}$ are large prime numbers. The CA makes N and e public. The CA also outputs a collision resistant cryptographic hash function H. Individuals looking for a pair also create RSA key pair (e,d) and make e public. The individuals further choose a username and an ID. Each individual's ID is the hash of his/her RSA private key. Let the attributes of Alice and the other j individuals looking for a match-pair be $a_i=\{a_1,a_2,\ldots,a_n\}$ and $b_{jk}=\{b_{j1},b_{j2},\ldots,b_{jp}\}$ respectively. Each person in the protocol then exponentiates the personal attributes using the public key of the CA. Alice's attributes hence becomes $a_i^e=\{a_1^e,a_2^e,\ldots,a_n^e\}$, whist the attributes of the other individuals become $b_{jk}^e=\{b_{j1}^e,b_{j2}^e,\ldots,b_{jp}^e\}$. Alice, and the other individuals, then encrypt their attributes, ID, username, and the public key pair of his/her RSA key using the public key of the CA and send it to the CA. Thus, Alice sends $E_e\{a_i^e \,\|\, ID_A \,\|\, username \,\|\, RSA_{publickey}, e_A\}$ to the CA. The other individuals also send $E_e\{b_{jk}^e \,\|\, ID_j \,\|\, username \,\|\, RSA_{publickey}, e_j\}$ to the CA.

When the attributes of Alice are received, the CA certifies them and returns $A=\{(a_1,s_1),(a_2,s_2),\ldots,(a_n,s_n)\}$; where $s_i=H(a_i)^d \bmod N$. Likewise, for the attributes of each of the other individuals, the CA certifies them and returns $B_{jk}=\{(b_{j1},\sigma_{j1}),(b_{j2},\sigma_{j2}),\ldots,(b_{jp},\sigma_{jp})\}$; where $\sigma_{jk}=H(b_{jk})^d \bmod N.$

This process is done just once for each member in the protocol. In the event that an individual wants to update the interests, s/he goes through the same process again. But before a new certificate is issued, the CA checks if there are any complaints about the individual. In the event that there are complaints against the individual, the CA refuses to renew the certificate.

3.2 Matchmaking Phase

Alice has private input $A=\{(a_1,s_1),(a_2,s_2),\ldots,(a_n,s_n)\}$; and the other individuals also have private input $B_{jk}=\{(b_{j1},\sigma_{j1}),(b_{j2},\sigma_{j2}),\ldots,(b_{jp},\sigma_{jp})\}$; . Alice chooses a random odd number $R_A \leftarrow_r Z_N$ and calculates $M_{A:i}=s_i^{R_A} \bmod N$, for all $i=1,\ldots,n$. Alice then sends $M_{A:i}, i=1,\ldots,n$ to the other individuals. Thus Alice sends $M_{A:i} \,\|\, M_{A:2} \,\|\, \ldots \,\|\, M_{A:n}$ to each of the other individuals. Also, each individual chooses a random odd number $R_{B_j} \leftarrow_r Z_N$ and calculates $M_{B_{jk}}=\sigma_{jk}^{R_{B_j}} \bmod N$, for all $j=1,\ldots,m$ and $k=1,\ldots,p$. Each individual then sends $M_{B_{jk}}, j=1,\ldots,m; k=1,\ldots,p$ to Alice. Thus each individual sends $M_{B_{j1}} \,\|\, M_{B_{j2}} \,\|\, \ldots \,\|\, M_{B_{jk}}$ to Alice. For all

$j=1,\ldots,m$ and $k=1,\ldots,p$, Alice computes $M'_{B:jk}=\left(M_{B:jk}\right)^{eR_A} \bmod N$ and sends $\mathrm{M}'_{B:jk}$, $j=1,\ldots,m; k=1,\ldots,p$ to each individual. Also, for all $i=1,\ldots,n$ each individual computes $M'_{A:i}=\left(M_{A:i}\right)^{eR_{B_j}} \bmod N$ and sends $M'_{A:i}, i=1,\ldots,n$ to Alice. In steps 7 and 8, Alice outputs $|I_A|$, the number of common attributes she has with the other individuals. Also, each of the other individuals outputs $|I_j|$, the number of common attributes each individual has with Alice. Alice then checks which individual's $|I_j|$ is at least the threshold number of attributes set by her. Hence, the individual(s) with $|I_j| \geq A_{Threshold}$ then becomes the matching-pair of Alice. At this point, Alice and the individual(s) in the protocol know only the number of attributes they have in common.

For simplicity, let us assume Bob was the only individual in the protocol that has the number of attributes that is at least $A_{Threshold}$. In order to know the actual attributes Alice and Bob have in common, they have to exchange their random numbers in algorithm 2. Alice and Bob undertake authenticated Diffie-Hellman protocol [14] to exchange their private keys (the random odd numbers in algorithm 1). After this exchange, secure communication can be undertaken.

3.3 Algorithms

In algorithm 1, Alice is able to find an individual(s) who has at least the threshold number of common attributes with her. At the end of this protocol, Alice and Bob will know only the number of attributes they have in common. Alice and Bob use algorithm 2 to establish a secure communication channel between them. Alice and Bob agree on a generator g. Both Alice and Bob using their random odd numbers in algorithm 1, undertake an authenticated Diffie-Hellman protocol. Hence, in steps 4 and 5 in algorithm 2, Alice sends R_A to Bob and Bob also sends R_B to Alice. After Alice has received $\left(g^{R_A}\right)^{R_B}$ she then computes to know R_B. Bob also after receiving $\left(g^{R_B}\right)^{R_A}$ computes to know R_A. At this point, they will be able to compute to know the actual attributes they have in common.

4. Security

The private sets of Alice and the individual(s) are certified by the CA. As a result, only certified set of attributes are used in the protocol. The certification of attributes by the CA in this protocol prevents a malicious person from cheating by changing his/her input attributes and to ensure that the attributes used in the protocol by Alice and the individuals, they really possess them.

Algorithm 1: Protocol for Computing the Number of Common Attributes

Require: Let $\{N, e, H\}$ be inputs from the CA common to Alice and the other individual(s)

1: Private attributes of Alice is $\left(a_1, a_2, \ldots, a_n\right)$ after certification, Alice's private input set becomes $A=\left\{\left(a_1, s_1\right),\left(a_2, s_2\right),\ldots,\left(a_n, s_n\right)\right\}$ where $s_i=H\left(a_i\right)^d \bmod N$.

2: Private attributes of the other j individuals is $\left(b_{j1}, b_{j2}, \ldots, b_{jp}\right)$; after certification, their private input set becomes $B_{jk} = \left\{\left(b_{j1}, \sigma_{j1}\right), \left(b_{j2}, \sigma_{j2}\right), \ldots, \left(b_{jp}, \sigma_{jp}\right)\right\}$, where $\sigma_{jk} = H\left(b_{jk}\right)^d \bmod N.$

3: Alice chooses a random odd number $R_A \leftarrow_r Z_N$ and calculates $M_{A:i} = s_i^{R_A} \bmod N.$ Alice then sends $M_{A:i}, i = 1, \ldots, n$ to each individual.

4: For all $j = 1, \ldots, m$ and $k = 1, \ldots, p,$ each individual chooses a random odd number $R_{B_j} \leftarrow_r Z_N$ and calculates $M_{B:jk} = \sigma_{jk}^{R_{B_j}} \bmod N.$

Each individual then sends $M_{B:jk},\ j = 1, \ldots, m$ and $k = 1, \ldots, p$ to Alice.

5: For all $j = 1, \ldots, m$ and $k = 1, \ldots, p,$ Alice calculates $M^{'}_{B:jk} = \left(M_{B:jk}\right)^{eR_A} \bmod N.$ Alice then sends $M^{'}_{B:jk},\ j = 1, \ldots, m$ and $k = 1, \ldots, p,$ to each individual.

6: Also, for all $i = 1, \ldots, n$, each individual calculates $M^{'}_{A:i} = M_{A:i}^{eR_{B_j}} \bmod N.$

Each individual sends $M^{'}_{A:i}, i = 1, \ldots, n$ to Alice.

7: Alice computes and outputs the intersection $| I_A | \in A \cap B_{jk}$ if $\exists\, i, j$ and k $s.t. M^{'}_{A:i} = M^{'}_{B:jk}.$

8: Each individual also computes and outputs his/her intersection set, $| I_j | \in A \cap B_j$ if $\exists\, i, j$ and k $s.t.\ M^{'}_{B:jk} = M^{'}_{A:i}.$

In algorithm 1, for all the attributes of Alice and the other individual(s), the CA computes $s_i = H\left(a_i\right)^d \bmod N$ and $\sigma_{jk} = H\left(b_{jk}\right)^d \bmod N$ respectively. This computation is to certify the input attributes of the persons in the protocol. By this, the attributes of Alice and the other individual(s) in the protocol are bound to them. Hence, they cannot change or modify their attributes so as to gain more information from the others. This facilitates the security of this protocol as individual(s) cannot input attributes they do not possess. Alice and the other individual(s) in each step in the algorithm ensure that the other cannot know the actual attributes they possess before the protocol ends. An individual may terminate the protocol before it ends if s/he is able to know the other person's personal attributes. After Bob has been found as having at least the threshold number of attributes in algorithm 1, algorithm 2 ensures that Alice and Bob exchange their random odd numbers so as to compute their common attributes. To ensure that their random odd numbers are exchanged safely, Alice and Bob undertake authenticated Diffie-Hellman protocol in algorithm 2.

Algorithm 2: Authenticated Diffie-Hellman Protocol for Exchanging the Random Numbers of the Matched-pair.

Require: Alice has a random odd number R_A, Bob also has a random odd number, R_B.

1: Alice using the generator g computes and sends $g^{R_A} = Enc\left(g^{R_A} \,||\, ID_A\right)$ and sends $g^{R_A} \,||\, g^{R_B} \,||\, Sign_{Bob}\left(g^{R_A} \,||\, g^{R_B} \,||\, ID_A\right)$ to Alice.

2: Bob using the generator g computes $g^{R_B} = Enc\left(g^{R_B} \,||\, ID_B\right)$ and sends $g^{R_A} \,||\, g^{R_B} \,||\, Sign_{Alice}\left(g^{R_A} \,||\, g^{R_B} \,||\, ID_B\right)$ to Alice.

3: Alice computes and sends $Sign_{Alice}\left(g^{R_A} \| g^{R_B} \| ID_A\right)$ to Bob.

4: Alice computes $\left(g^{R_A}\right)^{R_B}$ and Bob also computes $\left(g^{R_B}\right)^{R_A}$

In order to enhance privacy in the proposed protocol, each person in the protocol will only know the intersection set and his/her personal input. The intersection set, $|I_A|$ computed by Alice and the intersection set, $|I_j|$ computed by each individual in algorithm 1 contain only the number of common attributes but not the actual attributes. Apart from the intersection set, nothing else will be learnt from the protocol. Also, this proposed protocol guards against spoofing. Thus, the attributes of the persons in the protocol are authorized. Hence, a dataset containing a user's attributes can be queried by another user if the owner of the specific queried dataset authorizes the user. In other words, a user cannot generate a query without authorization from the dataset owner. In this protocol, we cannot absolutely prevent user profiling as the initiator and its best matching-pair will mutually learn their intersection set. Thus, our protocol seeks to minimize the amount of private information revealed in one protocol run. Our protocol is also collusion resistant as members cannot collude to know the attributes of Alice. The persons in the protocol cannot collude to know the attributes of Alice as they do not know the existence of the other persons in the protocol.

At the end of the protocol, Alice outputs, $M'_{A:i} = M'_{B:jk}$, where;

$$M'_{A:i} = \left(M_{A:i}\right)^{eR_{B_j}} \bmod N = \left(s_i^{R_A} \bmod N\right)^{eR_{B_j}} \bmod N = \left(s_i\right)^{eR_A R_{B_j}} \bmod N$$

$$= \left[H\left(a_i\right)^d\right]^{eR_A R_{B_j}} \bmod N = \left[H\left(a_i\right)\right]^{R_A R_{B_j}} \bmod N.$$

$$Also, M'_{B:\mathrm{jk}} = \left(M_{B:jk}\right)^{eR_A} \bmod N = \left(\sigma_{jk}^{R_{B_j}} \bmod N\right)^{eR_A} \bmod N = \left[H\left(b_{jk}\right)^d\right]^{eR_A R_{B_j}} \bmod N$$

$$= \left[H\left(b_{jk}\right)\right]^{R_A R_{B_j}} \bmod N.$$

Both $\left[H\left(a_i\right)\right]^{R_A R_{B_j}} \bmod N$ and $\left[H\left(b_{jk}\right)\right]^{R_A R_{B_j}} \bmod N$ give the number of common attributes Alice has with each individual.

Furthermore, at the end of the protocol, each individual outputs;

$M'_{B:jk} = M'_{A:i}$ where,

$$M'_{B:jk} = \left(M_{B:jk}\right)^{eR_A} \bmod N = \left(\sigma_{jk}^{R_{B_j}} \bmod N\right)^{eR_A} \bmod N = \left(\sigma_{jk}\right)^{eR_A R_{B_j}} \bmod N$$

$$= \left[H\left(b_{jk}\right)^d\right]^{eR_A R_{B_j}} \mathrm{mod} N = \left[H\left(b_{jk}\right)\right]^{R_A R_{B_j}} \mathrm{mod} N.$$

$$\text{Also,} M'_{A:i} = \left(M_{A:i}\right)^{eR_A R_{B_j}} \bmod N = \left(s_i^{R_A} \bmod N\right)^{eR_{B_j}} \bmod N = \left[H\left(a_i\right)^d\right]^{eR_A R_{B_j}} \bmod N$$

$$= \left[H\left(a_i\right)\right]^{R_A R_{B_j}} \bmod N.$$

Both $\left[H\left(b_{jk}\right)\right]^{R_A R_{B_j}} \bmod N$ and $\left[H\left(a_i\right)\right]^{R_A R_{B_j}} \mathrm{mod} N$ give the number of common attributes each individual has in common with Alice.

Thus the protocol is correct. Alice computes, $M^{'}_{A:i} = M^{'}_{B:jk}$ and it can be observed that, $\left[H\left(a_i\right)\right]^{R_A R_{B_j}} \bmod N = \left[H\left(b_{jk}\right)\right]^{R_A R_{B_j}} \bmod N$. Also, each individual computes $M^{'}_{B:jk} = M^{'}_{A:i}$ and it can be observed that, $\left[H\left(b_{jk}\right)\right]^{R_A R_{B_j}} \bmod N = \left[H\left(a_i\right)\right]^{R_A R_{B_j}} \bmod N$. The outputs of both Alice and each individual are same hence, the protocol is correct.

4.1 Achievement of Privacy Levels

Privacy level 1 is achieved in steps 7 and 8 of algorithm 1. In step 7, Alice computes and outputs $| I_A | \in A \cap B_{jk}$ if $\exists\ i, j$ and k s.t. $M^{'}_{A:i} = M^{'}_{B:jk}$. Hence, $| I_A |$ allows Alice to know only the number of attributes she has in common with an individual(s). In like manner, in step 8 each individual computes and outputs $| I_j | \in A \cap B_{jk}$ if $\exists\ i, j$ and k s.t. $M^{'}_{B:jk} = M^{'}_{A:i}$. The computation of $| I_j |$ allows each individual to know the number of attributes s/he has in common with Alice.

Privacy level 2 is achieved after completing algorithm 2. When they complete algorithm 2, each will know the random odd number of the other. With the knowledge of these random odd numbers, Alice and Bob will know the actual attributes they have in common with each other.

5. CONCLUSION

With the increasing popularity of mobile social networks, it is important to develop secure and privacy preserving protocols to enable users to effectively interact with each other. In our protocol, an individual can find the best match-pair from among many potential individuals by setting a threshold number of attributes that another individual should possess in order to qualify as a pair. This is executed without any other individual knowing any information about the matched-pair. Our protocol for matchmaking preserves users' information from unnecessary leakage of private and personal information.

References

[1] G. Ateniese, E. D. Cristoforo and G. Tsudik, "(If) size matters: size hiding private set intersection", in Public Key Cryptography, **(2011)**, pp. 156-173.

[2] L. Kissner and D. Song, "Privacy-preserving set operations", Advances in Cryptology-Crypto, LNCS, vol. 3621, **(2005)**, pp. 241-257.

[3] Y. Li, J. D. Tygar and J. M. Hellerstein, "in Computer Security in the 21st Century", chapter 3, Springer, New York, NY, USA, **(2005)**, pp. 25-50.

[4] E. De Cristofaro and G. Tsudik, "Practical private set intersection protocols with linear complexity", in Financial Cryptography and Data Security, **(2010)**, pp. 143-159.

[5] R. W. Baldwin and W. C. Gramlich, "Cryptographic Protocol for Trustable Match Making", IEEE Security and Privacy, **(1985)**, pp. 92-100.

[6] A. Pietilainen, E. Oliver, J. LeBrun, G. Varghese and C. Diot, "Mobiclique: middleware for mobile social networking", in Proceedings of the 2nd ACM Workshop on Online Social-networks, **(2009)**, pp. 49-54.

[7] E. De Cristofaro, Y. Lu and G. Tsudik, "Efficient techniques for Privacy-preserving sharing of sensitive information", International Conference on Trust and Trustworthy Computing (TRUST), **(2011)**, pp. 239-253.

[8] Y. Sang and H. Shen, "Privacy Preserving Set Intersection Protocol Secure Against Malicious Behaviors", Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies, **(2007)**, pp. 461-468.

[9] Z. Yang, B. Zhang, J. Dai, A. Champion, D. Xuan and D. Li, "Esmalltalker: A distributed mobile system for social networking In physical proximity", in IEEE, ICDCS, **(2010)**, pp. 468-477.

[10] 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, **(2013)**, pp. 2024-2033.

[11] N. Eagle and A. Pentland, “Social Serendipity: Mobilizing Social Software”, In IEEE Pervasive Computing, Special Issue: The Smartphone, **(2005)**, pp. 28-34.

[12] Q. Xie and U. Hengartner, “Privacy-Preserving Matchmaking for Mobile Social Networking Secure Against Malicious Users”, In Proc. 9th Int’l. Conf. on Privacy, Security (PST), and Trust, **(2011)**, pp. 252-259.

[13] R. Carthy, “Will IRSeek have a chilling effect on IRC chat? 2007 Also in A. Narayanan and V. Shmatikov”, Deanonymizing Social Network, 30th IEEE Symposium on Security and Privacy, **(2009)**.

[14] D. R. Stinson, “Cryptography Theory and Practice”, Third Edition, Chapman and Hall/CRC, Taylor and Francis Group, **(2006)**, pp. 431-435.

[15] R. Agrawal, A. Evfimievski and R. Srikant, “Information Sharing Across Private Databases”, in Proc. of SIGMOD, **(2003)**, pp. 86-97.

[16] J. Camenisch and G. M. Zaverucha, “Private intersection of certified sets”, In Financial Cryptography and Data Security, Springer, **(2009)**, pp. 108-127.

[17] Y. Wang, T. Zhang, H. Li, L. He and J. Peng, “Efficient Privacy Preserving Matchmaking for Mobile Social Networking against Malicious Users”, IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications, **(2012)**, pp. 609-615.

[18] E. De Cristofaro, S. Jarecki, J. Kim and G. Tsudik, “Privacy-preserving policy-based information transfer”, In Privacy Enhancing Technologies, **(2009)**, pp. 164-184.

[19] E. De Cristofaro, A. Durussel and I. Aad, “Reclaiming Privacy for Smartphone Applications”, In Proc. of Pervasive Computing and Communications (PerCom), IEEE International, **(2011)**, pp. 84-92.

[20] M. J. Freedman, K. Nissim and B. Pinkas, “Efficient private matching and set intersection”, Advances in Cryptology EUROCRYPT, **(2004)**, pp. 1-19.

[21] J. Kjeldskov and J. Paay, “Just-for-Us: A Context-Aware Mobile Information System Facilitating Sociality”, In Proc. 7th International. Conf. on Human Computer Interaction with Mobile Devices and Services, **(2005)**, pp. 23-30.

[22] K. Li, T. Sohn, S. Huang and W. Griswold, “PeopleTones: A System for the Detection and Notification of Buddy Proximity on Mobile Phones”, In Proc. 6th Intl. Conf. on Mobile Systems (MobiSys), **(2008)**, pp. 160-173.

[23] M. Li, N. Cao, S.Yu and W. Lou, “FindU: Privacy-preserving personal profile matching in mobile social networks”, In Proc. of IEEE Infocom, **(2011)**, pp. 2435-2443.

[24] C. Hazay and Y. Lindell, “Efficient Protocols for Set Intersection and Pattern Matching with Security Against Malicious and Covert Adversaries”, In Journal of Cryptology, vol. 23, no. 3, **(2010)**, pp. 422-456.

[25] S. Jarecki and X. Liu, “Efficient Oblivious Pseudorandndom Function with Applications to Adaptive OT and Secure Computation of Set Intersection”, In TCC, **(2009)**, pp. 577-594.

[26] R. Lu, X. Lin and X. (Sherman) Shen, “SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-health emergency”, IEEE transactions on parallel and distributed systems, vol. 24, no. 3, **(2013)**, pp. 614-624.

[27] D. Dachman-Soled, T. Malkin, M. Raykova and M. Yung, “Efficient robust private set intersection”, In LNCS, **(2009)**, pp. 125-142.

[28] “MeetGatsby”, meetgatsby, **(2011)** March, http://meetgatsby.com/, (Date assessed: 2014/03/20).

[29] Loopt, loopt, **(2011)** March, http://en.wikipedia.org/wiki/Loopt, (Date assessed: 2014/03/20).

[30] M. Rabin, “How to exchange secrets by oblivious transfer”, Tech. Rep. TR-81, Harvard Aiken Computation Laboratory, **(1981)**.

[31] A. Shamir, “Identity-based cryptosystems and signature schemes”, in Advances in Cryptology, Springer, Berlin, Germany, **(1985)**, pp. 47-53.

[32] 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.

[33] 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)**, http://eprint.iacr.org/ 2011/445.pdf.

[34] 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, **(2010)**, pp. 1-9.

[35] R. Lu, X. Lin, X. Liang and X. Shen, “Secure handshake with symptoms-macthing: the essential to the success of mhealthcare social network”, In Proc. BodyNets, Corfu Island, Greece, **(2010)**, pp. 683-694.

[36] S. Sarpong and C. Xu, “A Secure and Efficient Privacy-preserving Matchmaking for Mobile Social Network”, International Conference on Computer, Network Security and Communication Engineering, CNSCE, **(2014)**, pp. 362-366.

Authors

Solomon Sarpong, is a Ph.D. student in University of Electronic Science and Technology of China, Chengdu, (UESTC). His research interests include Information Security and Cryptography.

Chunxiang Xu, received her B.Sc., M.Sc. and Ph.D. degrees at Xidian University, P. R. China, in 1985, 1988, 2004 respectively. She is currently engaged in Information Security, Cloud Computing Security and Cryptography as a professor at University of Electronic Science and Technology of China, Chengdu, (UESTC).