IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 2 Issue 6, June 2015.
www.ijiset.com
ISSN 2348 – 7968

A collusion-resistant Privacy-preserving Attribute Matchmaking for Mobile Social Networks

Solomon Sarpong[1] and Chunxiang Xu[2]
Department of Computer Science University of Electronic Science and Technology of China Chengdu, China[1,2].

Abstract: Social networking is increasingly gaining popularity in recent times due to its flexibility and versatility. The recent advancements and improvements in appearance and functionalities of smart phones have made matchmaking on mobile social networks gain a lot of interest in recent times. Most users are more likely to use their mobile devices for social network activity like matchmaking than their computers or laptops. As a result, there is the need for secure protocols to protect users' information during the matchmaking. Hence, this research paper seeks to propose a secured and privacy-preserving protocol to protect users' information during matchmaking. In this proposed protocol, an individual becomes the matching-pair of the initiator if s/he has at least the threshold number of attributes pre-set by the initiator. Apart from protecting users' information from leaking, our proposed protocol is secured against malicious and semi-malicious attacks. Furthermore only the match-pair that has enough attributes to be a good pair gets to know the actual attributes they have in common.

Keywords: concatenation, privacy-preserving, nonspoofability, *k*-nearest neighbours, scalar-product computation.

I. INTRODUCTION

Mobile social network (MSN) has gone through very significant improvements since its inception and has gained a lot of popularity among users. This gain in popularity of MSN has necessitated the need for researchers to find more improved and diverse ways to help users feel secured and comfortable using them. Social network users tend to have a high level of trust toward other social network users. This trust can easily be abused by malicious users hence, researchers are coming up with protocols that are adequately secure to protect users' information. On MSN, a user's profiles usually contain sensitive information such as; user's full name, contact information, relationship status, date of birth, previous and current work, education background, private pictures e.t.c. This information hence makes such users' profiles prone to attacks.

The dramatic change in appearance and functionalities of smart phones has made matchmaking on mobile social networks gain a lot of interest in recent times. With this pervasive adoption of smart phones, there is a growing tendency to access our social networks more often by smart phones than desktop computers or laptops [46]. The hardware specifications of smart phones have been dramatically improved to the level of personal computers, along with friendly interface improvements and usability enhancements. WiFi and Bluetooth techniques are pervasively used in mobile applications to enable users to communicate with their physically-close neighbours [36].

The increased use of mobile phones has had significant impact on the way people interact. Mobile phone users have difficulty knowing where and how their information is stored and who is authorized to use it. Therefore, protecting mobile phone users' data and increasing their confidence in data privacy has become a real challenge. A mobile operator usually has private and sensitive information on users hence; a breach of confidentiality can result in severe embarrassment, financial loss, and even litigation for a mobile operator. Also, a breach of integrity can be devastating. Thus, any security incident may cause important business impact to the mobile operator [43]. Various authors describe privacy of user information differently. In this paper, privacy will be defined as personal information protected from malicious users but available to certain authorized persons Bnnig and Cap [37].

In matchmaking on MSN, users usually want to know what they have in common without divulging any other information. The exchange/sharing of information should be such that, unauthorized people not should be privy to the content of the information being shared. This has necessitated the need for protocols that protect users' information from unauthorized use. To protect users' sensitive information from leaking during matchmaking, some researchers have proposed protocols based on private set intersection [22], [26], [39], [40] and authorized private set intersection [33], [41], [42].

Privacy-preserving attribute matching finds useful application in situations where two users want to know if they have some attributes in common without disclosing any other information. Thus the design goal of PPAM is to help two users exchange personal information while preserving the privacy of their personal information. Hence, this paper seeks to propose a novel hybrid privacy-preserving attribute matchmaking protocol that will help individuals undertake matchmaking without leaking any information. Our proposed protocol will help users find the intersection of their attributes without leaking any other information that is not in the intersection. The matchmaking protocol first helps

users find the number of attributes they have in common. If they do not have enough attributes in common, they then terminate the protocol. Thus, the users exchange their attributes only when the number of attributes they have in common is at least the threshold number set by the initiator.

In the rest of the paper, we take brief look at private matchmaking protocols and privacy-preserving scalar product computation. Related work and our matchmaking protocol are in sections II and III respectively. The security of our protocol is in section IV. We conclude the paper in section V.

Private matchmaking protocols

Generally speaking, private set intersection is a cryptographic protocol that involves two individuals, each with a private set of attributes. Their goal is to compute the intersection of their respective attributes, such that minimal information is leaked in the process. Private matchmaking as an aspect of private set intersection arises when individuals want to know their common attributes. This can be done by computing the intersection of their respective data items. Private set intersection protocols found [26], [27], [45] in are not appropriate enough to be used for matchmaking. In these protocols, the individuals in the matchmaking protocol can use any items even if they do not really have them. In order to prevent the individuals in the protocol from populating their dataset with items they do not possess, authorised private set intersection is used. In authorised private set intersection, the dataset items are authorised by a mutually trusted third party [16], [41], [42]. This characteristic prevents cheating hence making it appropriate for matchmaking.

There are private set intersection protocols that allow only one of the individuals to know the intersection whilst the other learns nothing. Other protocols also allow both individuals to learn the intersection set only but nothing else. Also, some of the protocols employ the services of a certification authority to compute the intersection. Other protocols use cryptographic techniques so that the intersections are computed by each of the individuals in the protocol. Protocols by Agrawal et. al. [22], Vaidya et. al. [23], Arb et. al. [24] and Li et. al. [25] uses commutative encryptions to achieve private set intersection in matchmaking. Also, applications of oblivious polynomial evaluation can be found in research by Freedman et. al. [26], Kissner and Song [27], Sang et. al. [28], Ye et. al. [29] and Dachman et. al. [30]. Furthermore, Hazay and Lindell [31], Jarecki and Liu [32], and Cristofaro and Tsudik [33] applied oblivious pseudo random functions to achieve private set intersection. It must be observed that these matchmaking protocols are asymmetric. Hence, when using these protocols in matchmaking, the individual that computes the intersection is expected to report to the other truthfully. The asymmetric nature of these protocols makes them not desirable for matchmaking as their usage can lead to information asymmetry.

Privacy-preserving Scalar Product Computation

Privacy-preserving scalar computation finds useful application when data are partitioned horizontally among different people, sites, servers or organizations. Thus, with the application of privacy-preserving scalar computation, common items in the various datasets can easily be identified without revealing the content of the datasets. The application of privacy-preserving scalar computation can help different security agencies within or without countries to arrest or prevent criminal activities. In a situation where only *k*-nearest neighbours (*K*NN) is used to classify an individual as dangerous or otherwise, an individual can easily be misclassified. This misclassification can have dire consequences. An individual maybe misclassified because in *K*NN, an individual is classified based on the nearest persons of the individual. Hence, in the situation where there are not enough persons nearest to that individual in the database, there will be misclassification. On the other hand with the usage of privacy-preserving scalar computation, adequate information on an individual can be sort from different security agencies. With the usage of privacy-preserving scalar computation, the different agencies need not reveal the content of their databases but adequate information can be sought on an individual.

Yang et. al. [1] observed that secure scalar product protocol is a type of specific secure multi-party computation problem. Using this kind of protocol, they observed that the two parties involved are able to jointly compute the scalar product of their private vectors, but no party will reveal any information about his/her private vector to another one. It has been observed that privacy-preserving scalar-product computation in data analysis has different applications: Amirbekyan and Estivill-Castro [2], Du et. al. [3], Du and Atallah [4] applied it in regression; Du et al. [3], Du and Atallah [4] used it in calculation of the product of matrices; in clustering, Amirbekyan and Estivill-Castro [5], Estivill-Castro [6] and decision trees classification in Du and Zhan [7]. In [8], secure scalar product protocol based on homomorphic encryption and permutation on vectors was used but it was observed that the add vector used can result in all the vector attributes being the same, hence leaking information.

One of the ways to find the number of common items in a horizontally partitioned data is the use of scalar product computation. In our protocol, in order to ensure the privacy of the users' attributes and prevent the problem in [8], binary vector of attributes is used in the scalar product computation. The binary vectors represent items that are present or not in the database. The naive way of computing scalar products is either to let a trusted authority do the

ISSN 2348 – 7968

computation or allow one of the parties do the computation. But it must be observed that, the individual private set of attributes may be very large. Goethals et. al. [34] using a provably private scalar product protocol that is based on homomorphic encryption proposed an improved and efficient protocol that it can also be used on massive datasets. Also, Melchor et. al. [35] discussed the application of scalar product to the privacy preserving computation of trust.

II. RELATED WORK

In matchmaking, the techniques in use include; the use of a trusted third party technique, the fully distributed technique and the hybrid technique − a combination of the two fore-mentioned techniques. In the use of a trusted third party technique, the trusted third party is involved in each step of the matchmaking. Hence, the trusted third party knows everything about the matchmaking – from the individual matchmaker's attributes to the attributes of the matched-pair, including location parameters. These techniques can be found in [10], [18], [19]. With the involvement of the trusted third party in every step of the matchmaking, appropriate security measures should be put in place so as the make it more secured to withstand malicious and semi-honest attacks. In Just-for-Us [18], the application keeps track of the user's location, current activity, the location and activities of other people, and the current environmental conditions. The application then notifies users of friends within his/her close proximity.

There is another matchmaking technique that requires no trusted third party. Attributes of the initiator and the other match-seekers are shared using Shamir secret sharing scheme. The application of this technique can be found in [9], [14], [15]. In FindU [15], there is no usage of a trusted third party. An initiating user can find from a group of users the one whose profile best matches with his/her. In their protocol, in order to limit the risk of privacy exposure, only necessary and minimal information about the private attributes of the participating users is exchanged. In this protocol, to enhance security, several increasing levels of user privacy are defined, with the corresponding decreasing amounts of exchanged profile information.

There is also the hybrid technique used in matchmaking. The trusted third party is needed in this protocol just to oversee it. Thus, the trusted third party is not involved in the matchmaking protocol. This matchmaking technique can be found in [11], [12], [13], [17], [20], [21], [44]. In Wang et. al. [12], the initiator looks for a match-pair from other potential candidates using the hybrid technique of matchmaking. A match-pair is made with a candidate that has the greatest number of common attributes with the initiator. In this protocol, the computation of the intersection of the attributes is done mutually. Furthermore, to enhance the security of this protocol, the CA verifies the content of the intersections computed by the pair. If the intersections set are the same, the CA notifies them that the matching was successful. Also, in [17] and [21] the proposed protocols ensure that a pair is made when a candidate has a minimum pre-set threshold number of common attributes with the initiator. Furthermore, the CA verifies that the content of the individual intersection computed are the same. The CA then notifies the match pair of a successful match if the content of the intersection are the same.

III. MATCHMAKING DESIGN

It has been observed that social network users tend to have a high level of trust toward other social network users hence, this protocol will provide adequate security protection for users' information. Hence, this matchmaking protocol seeks to help a user find the best matching-pair among several other users(s). In this protocol, the best matching-pair is the user(s) that has attributes that is at least the pre-set threshold number of attributes of the initiator. The threshold number of attributes is the minimum number of attributes that a user(s) should possess so as to qualify as a match-pair. In our protocol, we assume that; (i). The CA cannot be compromised; (ii). Users will not terminate the protocol before it ends once they have started; (iii). Users keep their private keys safe to prevent impersonation; (iv). Users know the size of their intersection sets mutually; (v). Only the matched-pair know the actual attributes they have in common; (vi). Attacks from persons outside the protocol are not possible. In order to achieve the necessary security in our protocol, these privacy levels are defined.

Privacy Level 1:
When algorithm 1 ends, each user will know the number of attributes s/he has in common with the other users mutually. The number of attributes a user has in common with other user(s) will be known only to both of them but to no one else. Apart from this information, no other attribute information will be known by the users.

Privacy Level 2:
At the end of algorithm 2, both users will know the intersection set between them mutually. Also, apart from this information, no other attributes information will be known by the users.

Privacy Level 3:
At the end of the matchmaking protocol, users will know the actual attributes they have in common mutually. This information will be known to only the matched-pair and no one else.

ISSN 2348 – 7968

Due to the importance and private nature of the information that is shared on MSN, our protocol will seek to achieve these levels of privacy. The procedure for our matchmaking protocol can be generally partitioned into two phases. In phase 1, each user sets a threshold number of common attributes that another user(s) should possess so as to qualify as a matching-pair. Each user performs privacy-preserving scalar computation with each user to verify which user has at least the pre-set minimum threshold number of attributes. A user(s) that satisfies this criterion in phase 1 then proceed to phase 2. In phase 2, the matched users then execute algorithm 2 in order to exchange their attributes. If it is observed that there is no semi-honest behaviour by any user, the CA then sends the random number and the random permutation **to** the match-pair. At the end of these phases, the matched pair will know the number and type of attributes they have in common. On the other hand, the other users who did not have enough attributes with each other to be match-paired will only know the number of attributes they have in common with each other.

Table 1: Notations

$y_{i_{Threshold}}$	Threshold number of attributes of user y_i
$\|\|$	Concatenation
ID_{y_i}	Identity of user y_i
$\| I_{Alice} \|$	Number of attributes in intersection computed by Alice
$\| I_{Bob} \|$	Number of attributes in intersection computed by Bob
φ	Random permutation
(e, d)	RSA-key pair of the CA
(e_{y_i}, d_{y_i})	RSA-key pair of user y_i

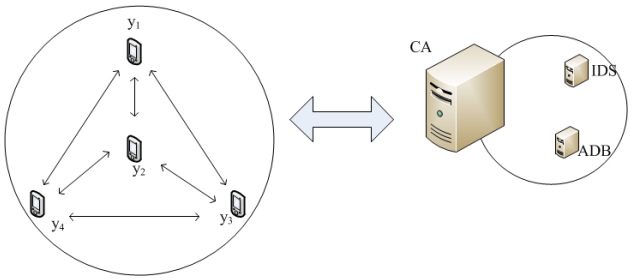
Matchmaking Architecture

Users: Each user creates an RSA key pair (e_{y_i}, d_{y_i}), chooses a username and then makes the username together with the e_{y_i} public. Let $Y = \{y_1, y_2, y_3, \ldots\}$ be the set of registered users. Each user then selects n attributes; let $A_i = \{a_{i1}, a_{i2}, \ldots, a_{in}\}$ be the set of attributes of user y_i. Each user is equipped with a mobile phone that has his/her attributes configured in it before the protocol begins. Users' mobile phones are within communication range of each other using Bluetooth or Wi-Fi. Each user sets a threshold number of attributes, $y_{i_{Threshold}}$, that another user should have in common with him/her so as to qualify as his/her match-pair.

The Certification Authority (CA): The CA comprises an Identity Signer (IDS) and Attributes Database (ADB). The identity signer chooses a random number, R; creates an RSA key pair (e, d) and makes e public. The identity signer stores the identity information of users. The Attribute Database has N attributes and stores information pertaining to users' attributes. The Attributes Database fixes the maximum number of attributes, $n (n << N)$ which user(s) can use in the matchmaking protocol. A user registers with the CA by sending the username and the public key pair of the RSA key, e_{y_i}, to the identity signer. The identity signer gives each user an identity, ID_{y_i}. Each user then sends $E_e[\{attributes\ of\ user\ y_i\} \| username_{y_i} \| ID_{y_i}]$ to the CA. After receiving this, the CA then forms a binary vector of attributes and returns $E_{e_{y_i}}\{ID_{y_i} \| binary\ vector\ of\ attributes\ of\ user, y_i\}$ to user, y_i. Hence, in a user's binary set of attributes, $a_i = 1$ if the attribute exits and $a_i = 0$ if otherwise.

Secure communication channels are established between users' devices by using their public/private key pair. The architecture for our matchmaking protocol is as depicted in figure 1. In figure 1, a node represents a social network user, and an edge represents relationship between two social network users. Each social network user has information ranging from demographic information, contact information, comments, pictures, videos, etc.

ISSN 2348 – 7968



Matchmaking Protocol

Using the binary vector of attributes, each user then undertakes privacy-preserving scalar computation as described in algorithm 1 with each registered user. This will enable each user know the number of attributes s/he has in common with other users. Privacy-preserving scalar computation between two users y_1 and y_2 results in $A_1 \times A_2^T = \sum_{i=1}^{N} a_{1i} \times a_{2i}$. This computation helps both y_1 and y_2 to know the number of attributes they have in common with each other mutually. User y_1 then checks if $A_1 \times A_2^T = \sum_{i=1}^{N} a_{1i} \times a_{2i}$ is at least the threshold number of common attributes, $y_{1_{Threshold}}$, set by him/her. Likewise, y_2 also checks if $A_1 \times A_2^T = \sum_{i=1}^{N} a_{1i} \times a_{2i}$ is at least the threshold number of common attributes, $y_{2_{Threshold}}$, set by him/her. This privacy-preserving scalar computation is undertaken by all pairs of $Y = \{y_1, y_2, y_3, \ldots\}$ users. Each user then checks if any other user has enough attributes with him/her so as to qualify as a match-pair. In this case, if user y_1 observes that y_2 has attributes that is at least the threshold set by him/her, y_2 becomes the match-pair of y_1.

Algorithm 1: Scalar product computation of the number attributes

Require: The CA has a vector of *N* attributes.

1: CA creates RSA key-pair (e, d) and makes e public.

2: Let the users be $Y = \{y_1, y_2, y_3, \ldots\}$. Each user y_i has $A_i = \{a_{i1}, a_{i2}, \ldots, a_{in}\}$ attributes, where $n << N$.

3: Each user chooses a username, creates an RSA key-pair and makes the RSA public-key e_{y_i} together with his/her username public. A user registers with the CA by sending the username and his/her RSA public key. The CA then returns to each user an identity, ID_{y_i}

4: Each user sets a threshold number of attributes, $y_{i_{Threshold}}$ which qualifies another user to be a pair. A user then sends $E_e[\{attributes\ of\ user\ y_i\} \| username_{y_i} \| ID_{y_i}]$ to the CA.

5: After receiving $E_e[\{attributes\ of\ user\ y_i\} \| username_{y_i} \| ID_{y_i}]$, the CA forms a binary vector of attributes. The CA then returns $E_{e_{y_i}}\{ID_{y_i} \| binary\ vector\ of\ attributes\ of\ user, y_i\}$ to each user, y_i. Hence, in user $y_i's$ binary vector, $a_i = 1$ if the attributes exits and $a_i = 0$ if otherwise.

6: User y_1 sends $E_{e_{y_i}}\{ID_{y_i} \| binary\ vector\ of\ attributes\ of\ user, y_1\}$ to users $y_i, i = 2, 3, \ldots$

Likewise, user y_2 sends $E_{e_{y_i}}\{ID_{y_i} \| binary\ vector\ of\ attributes\ of\ user, y_2\}$ to users $y_i, i = 1, 3, \ldots$ and so on.

ISSN 2348 – 7968

7: Each user performs privacy-preserving scalar computation with each of the other users. Users y_1 and y_2 perform privacy-preserving scalar computation $A_1 \times A_2^T = \sum_{i=1}^{N} a_{1i} \times a_{2i}$. This computation helps each user to know the number of attributes s/he has in common with each of the other users.
8: Each user then checks which other user(s) has number of attributes that is at least his/her pre-set threshold number of common attributes. Thus user y_1 verifies if $\sum_{i=1}^{N} a_{1i} \times a_{2i} \geq y_{1_{Threshold}}$. Likewise, user y_2 verifies if $\sum_{i=1}^{N} a_{1i} \times a_{2i} \geq y_{2_{Threshold}}$.

In our matchmaking protocol, attributes are the same if they are semantically the same. Let us assume users y_1 and y_2 represent Alice and Bob respectively and Alice is the initiator. Also, let us assume the number of attributes that Bob has in common with Alice is at least the threshold set by her. As Bob has the number of common attributes that is at least the threshold set by Alice, Bob becomes match-pair of Alice. They then proceed to execute algorithm 2. Alice sends $E_e\{ID_{Alice} \,||\, ID_{Bob} \,||\, \sum_{i=1}^{N} a_{1i} \times a_{2i}\}$ to the CA. Also, Bob sends $E_e\{ID_{Bob} \,||\, ID_{Alice} \,||\, \sum_{i=1}^{N} a_{1i} \times a_{2i}\}$ to the CA. The CA checks if $\sum_{i=1}^{N} a_{1i} \times a_{2i}$ from Alice and Bob are the same. If they are not the same, the algorithm is terminated and the CA checks who is involved in a semi-honest attack on the protocol. The user who is involved in the semi-honest attack is then removed from the list of registered users. If they are the same, the algorithm continues.

The CA then exponentiates the attributes of Alice and Bob using the random number, R. After the exponentiation of the individual attributes, the CA randomly permutes them using the random permutation, φ. The CA sends $E_{e_{Alice}}[\varphi\{a_{11}^R, \ldots, a_{1n}^R\}]$ and $E_{e_{Bob}}[\varphi\{a_{21}^R, \ldots, a_{2n}^R\}]$ to Alice and Bob respectively. After receiving $E_{e_{Alice}}[\varphi\{a_{11}^R, \ldots, a_{1n}^R\}]$ from the CA, Alice sends $E_{e_{Bob}}[\varphi\{a_{11}^R, \ldots, a_{1n}^R\}]$ to Bob. Bob then decrypts it to obtain $\varphi\{a_{11}^R, \ldots, a_{1n}^R\}$. Likewise, Bob also sends $E_{e_{Bob}}[\varphi\{a_{21}^R, \ldots, a_{2n}^R\}]$ to Alice. Alice also obtains $\varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ after decrypting $E_{e_{Alice}}[\varphi\{a_{21}^R, \ldots, a_{2n}^R\}]$. Alice then computes the intersection $|I_{Alice}| \in \varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$. Bob also computes the intersection $|I_{Bob}| \in \varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$. The computation of the intersections $|I_{Alice}|$ and $|I_{Bob}|$ help both Alice and Bob respectively know the exponentiated form of the attributes they have in common. Alice sends $E_e\{ID_{Alice} \,||\, ID_{Bob} \,|||\, I_{Alice}|\}$ to the CA. Likewise, Bob also sends $E_e\{ID_{Bob} \,||\, ID_{Alice} \,|||\, I_{Bob}|\}$ to the CA. The CA checks if $|I_{Alice}|$ from Alice is the same as $|I_{Bob}|$ received from Bob. If $|I_{Alice}|$ and $|I_{Bob}|$ are the same, the CA sends the random permutation, φ and the random number, R to Alice and Bob. Hence, with the knowledge of φ and R, Alice and Bob can then retrieve the actual attributes they have in common from the intersection they computed. On the other hand, if $|I_{Alice}|$ and $|I_{Bob}|$ are not the same, the CA checks which user is involved in a semi-honest attack on the protocol. After the cheat is found out, the CA does not send φ and R to him/her. Hence, the user involved in the semi-honest attack cannot know the actual attributes s/he has in common with the pair. Furthermore, the cheat is removed from the list of registered users by the CA.

Algorithm 2: Exchanging of common attributes

Require: The CA chooses a random number $R \leftarrow_r Z_N$

1: Alice sends $E_e\{ID_{Alice} \,||\, ID_{Bob} \,|||\, I_{Alice}|\}$ to the CA. Likewise, Bob also sends $E_e\{ID_{Bob} \,||\, ID_{Alice} \,|||\, I_{Bob}|\}$ to the CA.

ISSN 2348 – 7968

2: The CA checks if $\sum_{i=1}^{N} a_{1i} \times a_{2i}$ from Alice is the same as that from Bob. If they are not the same, the algorithm is terminated and the CA checks who might have cheated. If they are the same, the algorithm continues.

3: Using the random number R, the CA exponentiates and randomly permutes the attributes of Alice to obtain $\varphi\{a_{11}^R, \ldots, a_{1n}^R\}$. The CA sends $E_{e_{Alice}}[\varphi\{a_{11}^R, \ldots, a_{1n}^R\}]$ to Alice.

4: Also, using the random number R, the CA exponentiates and randomly permutes the attributes of Bob to obtain $\varphi\{a_{21}^R, \ldots, a_{2n}^R\}$. The CA then sends $E_{e_{Bob}}[\varphi\{a_{21}^R, \ldots, a_{2n}^R\}]$ to Bob.

5: Alice sends $E_{e_{Bob}}[\varphi\{a_{11}^R, \ldots, a_{1n}^R\}]$ to Bob. Also, Bob sends $E_{e_{Alice}}[\varphi\{a_{21}^R, \ldots, a_{2n}^R\}]$ to Alice.

6: Alice obtains $\varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ from $E_{e_{Alice}}[\varphi\{a_{21}^R, \ldots, a_{2n}^R\}]$ and computes the intersection $|I_{Alice}| \in \varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$. This gives Alice the number of common attributes she has with Bob.

7: Bob also obtains from $\varphi\{a_{11}^R, \ldots, a_{1n}^R\}$ from $E_{e_{Alice}}[\varphi\{a_{11}^R, \ldots, a_{1n}^R\}]$ and computes the intersection $|I_{Bob}| \in \varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$. This gives Bob the number of common attributes she has with Alice.

IV SECURITY

Each user registers with the CA by sending $E_e[\{attributes\ of\ user\ y_i\} || username_{y_i} || ID_{y_i}]$ to the CA. The registration of the users with the CA ensures that users cannot modify their attributes so as to gain more information from other users. Hence, the registration binds a user's attributes to him/her. In step 5 of algorithm 1, the CA returns a binary vector of users' attributes to the users. The binary vectors of attributes are used by the users to compute the number of attributes each user has in common with the other. Hence, a user in possession of the other users' binary vector of attributes will not be able to know the actual attributes of the others. At the end of algorithm 1, the privacy-preserving scalar computation helps each user to know only the number of attributes s/he has in common with each user.

Before Alice and Bob proceed to execute algorithm 2, the CA ensures that $E_e\{ID_{Alice} || ID_{Bob} || \sum_{i=1}^{N} a_{1i} \times a_{2i}\}$ and $E_e\{ID_{Bob} || ID_{Alice} || \sum_{i=1}^{N} a_{1i} \times a_{2i}\}$ received from Alice and Bob respectively have the same $\sum_{i=1}^{N} a_{1i} \times a_{2i}$. If $\sum_{i=1}^{N} a_{1i} \times a_{2i}$ from both Alice and Bob are not the same, the algorithm is terminated. The CA then removes the cheat from the list of registered users.

The CA sends randomized exponentiated attributes, $E_{e_{Bob}}[\varphi\{a_{11}^R, \ldots, a_{1n}^R\}]$ and $E_{e_{Alice}}[\varphi\{a_{21}^R, \ldots, a_{2n}^R\}]$ to Alice and Bob respectively. The exponentiation and randomization of the attributes prevent Alice from mapping her attributes in $\varphi\{a_{11}^R, \ldots, a_{1n}^R\}$ to the corresponding attributes in $A_1 = \{a_{11}, a_{12}, \ldots, a_{1n}\}$ in polynomial time. Also, when Alice sends $\varphi\{a_{11}^R, \ldots, a_{1n}^R\}$ to Bob, he cannot map the actual attributes a_{1i} to the corresponding attribute a_{1i}^R in polynomial time. In like manner, the exponentiation and randomization of the attributes prevents Bob from mapping his attributes in $\varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ to the corresponding attributes in $A_2 = \{a_{21}, a_{22}, \ldots, a_{2n}\}$ in polynomial time. Also, when Bob sends $\varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ to Alice, she cannot map the actual attributes a_{2i} to the corresponding attribute a_{2i}^R in polynomial time.

The computation of the intersection $|I_{Alice}| \in \varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ helps Alice to know the exponentiated attributes she has in common with Bob. Also, the computation of the intersection $|I_{Bob}| \in \varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ by Bob helps him know the exponentiated attributes he has in common with Alice. From this intersection, neither Alice nor Bob can know the actual attributes s/he has in common with each other in polynomial time.

ISSN 2348 – 7968

As a further check against semi-honest attacks, Alice and Bob send $E_e\{ID_{Alice} \,||\, ID_{Bob} \,||| \, I_{Alice} \,|\}$ and $E_e\{ID_{Bob} \,||\, ID_{Alice} \,||| \, I_{Bob} \,|\}$ respectively to the CA. The CA then checks if $|\, I_{Alice} \,|$ from Alice is the same as $|\, I_{Bob} \,|$ from Bob. If they are the same, the CA then sends the random permutation φ and the random number R to Alice and Bob. With the knowledge of φ and R, Alice and Bob will be able to compute and know the actual attributes they have in common. On the other hand, if $|\, I_{Alice} \,|$ and $|\, I_{Bob} \,|$ form Alice and Bob respectively are not the same, then the CA checks which user might have launched a semi-honest attack on the protocol. After the user who launched the semi-honest attack is found out, the CA does not send φ and R to that user. Hence, the cheat cannot compute to know the actual attributes s/he has in common with the match-pair. The CA removes the cheating user from the list of registered users.

Achievement of Privacy Levels

Privacy level 1: The computation of $A_1 \times A_2^T = \sum_{i=1}^{N} a_{1i} \times a_{2i}$ by Alice and Bob enables them know the number of attributes each has in common with the other. The knowledge of the number of attributes is exclusive to only the users (Alice and Bob) that did the computation and no one else.

Privacy level 2: At the end of algorithm 2, Alice computes $|\, I_{Alice} \,|$ and Bob computes $|\, I_{Bob} \,|$. These computations help both Alice and Bob know the exponentiated attributes they have in common with each other. Hence, the computation of $|\, I_{Alice} \,|$ and $|\, I_{Bob} \,|$ help Alice and Bob know the exponentiated attributes mutually. Also, the knowledge of $|\, I_{Alice} \,|$ and $|\, I_{Bob} \,|$ is exclusive only to Alice and Bob but to no one else.

Privacy level 3: This privacy level is achieved after the CA sends φ and R to Alice and Bob. With the knowledge of φ and R , Alice and Bob can retrieve the actual attributes from the intersections $|\, I_{Alice} \,| \in \varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ and $|\, I_{Bob} \,| \in \varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ they computed respectively. Hence, only Alice and Bob will know the type of attributes they have in common.

Correctness of our protocol

In algorithm 1, the privacy-preserving scalar computation of $A_1 \times A_2^T = \sum_{i=1}^{N} a_{1i} \times a_{2i}$ yields the same number of attributes for each pair of users. Thus, if Alice has attributes $A_1 = \{a_{11}, a_{12}, \ldots, a_{1n}\}$ and user Bob has attributes $A_2 = \{a_{21}, a_{22}, \ldots, a_{2n}\}$. The scalar product computation by Alice yields $A_1 \times A_2^T = \sum_{i=1}^{N} a_{1i} \times a_{2i}$. Also, the scalar product computation by Bob yields $A_1 \times A_2^T = \sum_{i=1}^{N} a_{1i} \times a_{2i}$. Since the scalar computations by both Alice and Bob yield the same result, algorithm 1 is correct.

In algorithm 2, Alice will be in possession of Bob's attributes in the form of $\varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ and Bob will also be in possession of Alice's attributes in the form of $\varphi\{a_{11}^R, \ldots, a_{1n}^R\}$. Alice computes the intersection $\varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ and Bob also computes the intersection $\varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$. Since the outcome of $\varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ from Alice and the outcome of $\varphi\{a_{11}^R, \ldots, a_{1n}^R\} \cap \varphi\{a_{21}^R, \ldots, a_{2n}^R\}$ from Bob are the same, algorithm 2 is correct.

Collusion Resistance

There also exists a potential attack from users who would like to know the private set of attributes of the CA. Hence, since the CA cannot be compromised through malicious attacks, a semi-malicious attack like collusion can be undertaken. With the maximum number of attributes n that each user should use in the matchmaking set by the CA, and the size of the CA's private set of attributes as N , the colluding adversaries should be at least $(N/n-1)$, Wang et.

ISSN 2348 – 7968

al. [44]. That is, at least this number of adversaries is needed to know the private set of attributes of the CA. As the private set of attributes of the CA is very large, it makes it very unlikely that there can be at least as many as $(N/n-1)$ malicious users who would like to know the attributes of the CA. As a result, the protocol will secure against collusion attacks.

V CONCLUSION

In this paper, we have presented a matchmaking protocol in mobile social network. This mobile matchmaking protocol preserves users' attributes from unnecessary leaks. This is partly because, only the matched pair that get to know the number of attributes they have in common. This protocol is also secured against malicious and semi-malicious attacks.

REFERENCES

[1] Yang B, Yu Y, Yang C-H. A secure scalar product protocol against malicious adversaries. Journal of Computer Science and Technology 28(1), 2013, pp. 152-158.
[2] Amirbekyan, A. and Estivill-Castro, V. (2007). Privacy preserving regression algorithms. The 3rd WSEAS International Symposium on Data Mining and Intelligent Information Processing, ISDM 2007, Beijing, China, pp. 37-45.
[3] Du, W., Han, Y.-S. and Chen, S. (2004). Privacy-preserving multivariate statistical analysis: Linear regression and classification, In Proc. of the SIAM International Conference on Data Mining (SDM), Florida, 2004..
[4] Du, W. and Atallah, M. (2001), Privacy-preserving cooperative statistical analysis, in Proceedings of the 17th Annual Computer Security Applications Conference (ACSAC), ACM SIGSAC, IEEE Computer Society, New Orleans, Louisiana, pp. 102-110.
[5] Amirbekyan, A. and Estivill-Castro, V. (2006), Privacy preserving DBSCAN for vertically partitioned data. IEEE International Conference on Intelligence and Security Informatics, ISI 2006, Springer Verlag LNCS 3975, San Diego, CA, USA, pp. 141-153.
[6] Estivill-Castro, V. (2004), Private representative based clustering for vertically partitioned data, in R. Baeza-Yates, J. Marroquin and E. Chavez, eds, Fifth Mexican International Conference on Computer science (ENC 04), SMCC, IEEE Computer Society Press, Colima, Mexico, pp. 160-167.
[7] Du, W. and Zhan, Z. (2002), Building decision tree classifier on private data, in V. Estivill-Castro and C. Clifton, eds, Privacy, Security and Data Mining, IEEE ICDM Workshop Proceedings, Vol. 14 in the Conferences in Research and Practice in Information Technology Series, Australian Computer Society, Sydney, Australia, pp. 1-8.
[8] Amirbekyan A. and Estivill-Castro V. A new efficient privacypreserving scalar product protocol. In Proc. the 6th Australasian Data Mining Conference, Dec. 2007, pp. 209-214.
[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] N. Eagle and A. Pentland. Social Serendipity: Mobilizing Social Software. In IEEE Pervasive Computing, Special Issue: The Smartphone, 2005, pp. 28-34.
[11] Q. Xie, and U. Hengartner. Privacy-Preserving Matchmaking for Mobile Social Networking Secure Against Malicious Users. In Proc. 9th Int0l. Conf. on Privacy, Security (PST), and Trust 2011, pp. 252-259.
[12] 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.
[13] 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.
[14] 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, 2013, 24(3), pp. 614-624.
[15] M. Liu and W. Lou. FindU: Privacy-preserving personal profile matching in mobile social networks. In Proc. of IEEE Infocom, 2011, pp. 24352443.
[16] E. De Cristofaro and G. Tsudik. Practical private set intersection protocols with linear complexity. In Financial Cryptography and Data Security, 2010, pp. 143-159.

ISSN 2348 – 7968

[17] 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, 2014, (CNSCE, 2014), pp. 362-366.
[18] 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.
[19] K. Li, T. Sohn, S. Huang, 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.
[20] 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, 2013, 12(5), pp. 2024-2033.
[21] S. Sarpong and C. Xu. A Secure and efficient privacy-preserving attribute matchmaking protocol in proximity-based mobile social networks. 10th InternationalConference, Advance data Mining and Applications, Guilin (ADMA, 2014),LNCS Vol. 8933, pp. 305-318.
[22] R. Agrawal, A. Evffmievski, and R. Srikant, Information sharing across private databases, in SIGMOD 03. New York, NY, USA: ACM, 2003, pp. 86-97.
[23] J. Vaidya and C. Clifton, Secure set intersection cardinality with application to association rule mining, J. Comput. Secur., 13(4), 2005, pp. 593-622.
[24] M. von Arb, M. Bader, M. Kuhn, and R. Wattenhofer, Veneta: Serverless friend-of-friend detection in mobile social networking, in IEEE WIMOB 2008, pp. 184-189.
[25] Y. Li, J. Tygar, and J. Hellerstein, Private matching, in Computer Security in the 21st Century, 2005, pp. 25-50.
[26] M. Freedman, K. Nissim, and B. Pinkas, Efficient private matching and set intersection, in EUROCRYPT04. Springer-Verlag, 2004, pp. 1-19.
[27] L. Kissner and D. Song, Privacy-preserving set operations, in CRYPTO 05, LNCS. Springer, 2005, pp. 241-257.
[28] Y. Sang, H. Shen, and N. Xiong, Efficient protocols for privacy preserving matching against distributed datasets, in ICICS 06. Springer-Verlag, 2006, pp. 210-227.
[29] Q. Ye, H. Wang, and J. Pieprzyk, Distributed private matching and set operations, in ISPEC 2008, pp. 347-360.
[30] D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung, Efficient robust private set intersection, in ACNS 2009, pp. 125-142.
[31] C. Hazay and Y. Lindell, Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries, in TCC, 2008, pp. 155-175.
[32] S. Jarecki and X. Liu, Efficicient oblivious pseudorandom function with applications to adaptive ot and secure computation of set intersection, in TCC 2009. Berlin, Heidelberg: Springer-Verlag, pp. 577-594.
33. E. D. Cristofaro and G. Tsudik, Practical private set intersection protocols with linear complexity, in Financial Cryptography and Data Security, 2010.
[34] Goethals B., Laur S., Lipmaa H., and Mielikainen T. On Private Scalar Product Computation for Privacy-Preserving Data Mining.
[35] Melchor A. C., Ait-Salem B., and Gaborit P. A Collusion-Resistant Distributed Scalar Product Protocol with Application to Privacy-Preserving Computation of Trust. 2009 Eighth IEEE International Symposium on Network Computing and Applications.
[36] Liang X., Lu R., Lin X., and X. (Sherman). Security and Privacy in Mobile Social Networks, Springer Briefs in Computer Science, Springer New York Heidelberg Dordrecht London, 2013, pp. 1 - 9.
[37] Bunnig, C. Cap, C.H., Ad Hoc Privacy Management in Ubiquitous Computing Environments, Advances in Human-oriented and Personalized Mechanisms, Technologies, and Services, 2009. CENTRIC '09. Second International Conference on, 85(90), pp. 20-25.
[38] E. De Cristofaro and Y. Lu and G. Tsudik. Efficient techniques for privacy-preserving sharing of sensitive information. International Comference on Trust and Trustworthy Computing (TRUST), 2011, PP. 239-253.
[39] G. Ateniese, E. D. Cristoforo and G. Tsudik. (If) size matters: size hiding private set intersection. In Public Key Cryptography, 2011, pp. 156-173.
[40] Y. Sang, and H. Shen. Privacy Preserving Set Intersection Protocol Secure Against Malicious Behaviours. Eighth International Conference on Parallel and Distributed Computing, Applications and Technologies, 2007, pp. 461-468.
[41] J. Camenisch and G. M. Zaverucha. Private intersection of certified sets. In Financial Cryptography and Data Security, Springer, 2009, pp. 108-127.
[42] 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.
[43] C.K. Dimitriadis, Security for Mobile Operators in Practice, International Journal of Network Security, pp. 397-404, 2013.

ISSN 2348 – 7968

[44] Wang Y., Hou J., Tan Y-W., Nie X. A Recommendation-based Matchmaking Scheme for Multiple Mobile Social Networks against Private Data Leakage. Information Technology and Quantitative Management, ITQM 2013. Procedia Computer Science 17 (2013), pp. 781 - 788.
[45] G. Ateniese, E. D. Cristoforo and G. Tsudik. (If) size matters: size hiding private set intersection. In Public Key Cryptography, 2011, pp. 156-173.
[46] M. Stanley, Tablet demand and disruption mobile users come of age, Tech. Rep., 2011. www.morganstanley.com/views/perspectives/tablets-demand.pdf (Date accessed: 2015/02/19)