PPAM: Privacy-preserving Attributes Matchmaking Protocol for Mobile Social Networks Secure against Malicious Users

Solomon Sarpong, Chunxiang Xu, and Xiaojun Zhang
(Corresponding author: Solomon Sarpong)

Department of Computer Science, University of Electronic Science and Technology of China
Main Building A1-406, No. 2006, Xiyuan Avenue, West Hi-Tech Zone, Chengdu 611731, China
(Email: sarpong.uestc@gmail.com)
(Received June 12, 2015; revised and accepted Aug. 12 & Aug. 31, 2015)

Abstract

People often associate with others who share their hopes, aspirations, beliefs and experiences. This sense of belonging influences people when making friends be it physically or on social networks. Most of the existing matchmaking protocols just match-pair people without regards to the number of attributes they have in common. In lieu of these, we are proposing a hybrid matchmaking protocol that seeks to help match-pair seekers find the most appropriate pair. In our protocol, all private attributes are certified by a mutually trusted third party. Also, a candidate becomes a matching-pair of an initiator when s/he meets a criteria set by the initiator. In our protocol, the number of attributes in the intersection set is known mutually by both the initiator and the candidate but only the matched-pair gets to know the actual attributes they have in common. Furthermore, the protocol guards against malicious and semi-malicious attacks.

Keywords: Attributes, hybrid, matchmaking, nonspoofability

1 Introduction

1.1 Contextualization

In recent times, mobile telephony has changed the way we socialize and communicate. Currently, social networking has made a lot of gains in the cyberspace. The Internet has brought a new perspective about friends making and socializing. People no longer makes friends only in their neighborhoods and communities but also from all over the world.

The hardware specifications of smartphones have been dramatically improved to the level of personal computers along with friendly interfaces, improvements and usability enhancements. The smartphones have WiFi and Bluetooth interfaces that allow physically-close persons to communicate. Hence, with these improved features of smartphones, there is a growing tendency to access our social networks on our smartphones than on our desktop computers or laptops.

Furthermore, improvement in smartphones and people's eagerness to get information anytime-anywhere has increased the usage of Internet on mobile devices. This has brought the need for security of personal information. Data owners online have a problem with their data being used by unintended persons. Hence, the need to protect information of users has become very important. However, individuals can protect their own private or sensitive information by restricting the intended purpose of data access by denying the right to access for some purposes [7].

1.2 Relevance of the Theme

Matchmaking is a key component of mobile social networking [37]. In mobile social networking, persons form social networks based on a predefined criteria; for example former school mates, members of a club e.t.c. However, in a scenario where a person is looking for a recommendation, any individual(s) is not good enough but an individual(s) with specific qualities is appropriate. This is the premise of our research. The matchmaking protocol in this paper has an initiator looking for a person(s) with some particular characteristics to be his/her match-pair. Hence, the person(s) who qualifies to be a match-pair of the initiator should have a minimum number of attributes in common with the initiator.

1.3 Research Question

In matchmaking protocols, the matched-pair can terminate the protocol as a result of insufficient number of attributes they have in common. When the protocol is terminated, the individual's attributes would have been

known hence compromising the privacy and secrecy of the attributes. Hence, how can a pair know they have enough attributes in common before they exchange their attributes? This question has necessitated this research.

1.4 Objectives

Protocols for matchmaking such as [1, 3, 6, 12, 16, 17, 24, 37] simply match-pair the initiator and the candidate(s) without checking if they have enough attributes in common. The matched-pair at times terminate the protocol as a result of insufficient number of attributes they have in common compromising their attributes. However, protocols in [18, 19, 35, 36] sort to solve this problem by assuming that the candidate with the maximum intersection set with the initiator is the best matching-pair. We have realized that using this criteria is not good enough. Hence, as our contribution to research, we formulate matchmaking protocol that enables the initiator to check the number of attributes s/he has in common with a candidate before being match-paired. The initiator of the matchmaking sets a threshold number of attributes that candidate should have in common with him/her. If a candidate possesses at least these number of attributes, then the initiator and the candidate exchange their attributes. The novelty of our matchmaking protocol is that: (1) the initiator finds a match-pair that has at least the preset threshold number of attributes (2) the number of common attributes is known mutually by the persons in the protocol (3) the actual attributes are known only by the matched-pair in the protocol (4) the protocol can resist semi-honest and malicious attacks.

1.5 Limitations of the Paper

The matched-pair will know the actual attributes of the each other after they have exchanged them. Hence, a malicious person can do attribute profiling of the other persons s/he executed the protocol with. This protocol cannot prevent such a person from doing this. This is the main limitation of the protocol in this paper.

1.6 Structure of the Paper

The rest of this paper is organized as follows: we take a look at private set intersection in Section 2. In Section 3, we present related work. Our protocol, the algorithm for the matchmaking, the experimental implementation and the security of our algorithms are presented in Section 4. Finally, we conclude this paper in Section 5.

2 Private Set Intersection

When two persons want to find the common items in their individual private sets, they cannot just disclose the content of their sets so as to know the common items. This is what happens in matchmaking. Hence this brings the need for private set intersection protocols, PSI [1, 11, 13, 14, 15, 38]. PSI is a cryptographic protocol that allows two persons to compute the intersection of their private sets without disclosing any other information apart from what they have in common.

In PSI, the private inputs are chosen arbitrarily. This facilitates attacks from malicious users to gain extra information from other users [2]. In order to prevent this form of attack, authorized private set intersection, APSI [5, 10, 34] is used. In APSI, private inputs are certified by mutually trusted authority. Hence, in matchmaking protocols with variants of APSI, all the private input sets are certified by a mutually trusted party. The certification of attributes binds the private data sets to the data owners. This prevents the data owners from modifying their inputs so as to gain extra-information from other users. Certification of private data sets is important as no secure multi-party protocol can prevent a person in a protocol from cheating by changing his/her input before the protocol begins [2].

Certification of private inputs prevents malicious persons from claiming possession of fictitious data items in an attempt to find out if the other users possess those data items. In a distributed system framework, mutual authentication is becoming very important. Hence, it has become necessary for a user in such a system to verify the identity of the system or another user or node in the system verifies itself to him/her. Consequently, both user and the system may require some degree of authentication before information about them is released [9]. This mutual authentication is usually done by contacting a trusted third party. The use of a trusted third party may encounter the following challenges: (1) it may not be practical in a highly distributed system (2) sometimes the parties may not be willing to trust the third party (3) though the trusted third party may exist, it may not be available to all parties at all times. In light of all these problems, a solution researchers has developed is the use of cryptographic techniques. These cryptographic techniques enable users with private sets to verify whether or not their sets agree without revealing the content of their private sets.

3 Related Work

Baldwin and Gramlich [3] laid the foundation for matchmaking in social network with the use of a trusted third party protocol. Meldew [24] later proposed a protocol that did not rely on the use of a trusted third party. This protocol seemed to be more efficient than [3] as there was no need for trusted third party to be continuously available. In the matchmaking protocol proposed by Zhang and Needham [40], the matchmaking protocol depended on the availability of a public database service. Even though this protocol is efficient, the security of this protocol depends on the security of the hash function and the encryption algorithm used. Freedman, Nissim and Pinkas [38] also considered the problem of computing the

intersection of private data sets of two parties, where the data sets contain lists of elements taken from a large domain.

Shin and Gligor [32] observed that anonymity of protocol users, authentication of wishes and security in matchmaking was fundamental to private matchmaking. Hence, their proposed protocol sort to provide authentication for users and wish matches; privacy resistance to off-line dictionary attacks and forward privacy of users′ identities and their wishes. In their paper, Sang and Shen [25] addressed privacy preserving set intersection (PPSI) problem. Their paper sort to solve the problems associated with finding the intersection of data sets that are distributed on different sources while preserving the privacy of the data sets.

Shamir [31] proposed a scheme that enables any pair of users to communicate securely and to verify each other's signatures without exchanging private or public keys, without keeping key directories and without using the services of a third party. Camenisch et al. [4], proposed the searchable encryption scheme that provides an important mechanism to cryptographically protect data while keeping it available to be searched and accessed for matching information. In the scheme, they proposed two encryptions; public key encryptions with oblivious keyword search (PEOKS) and committed blind anonymous identity-based encryption. Lin et al. [21] proposed efficient blind-key encryption protocols for anonymous identity-based encryption and an anonymous hierarchical identity-based encryption. These schemes were used in privacy preserving profiles searching (PPPS) problem.

Sun et al. [33] proposed a privacy-preserving scheme for data sharing in social networks with efficient revocation for deterring a contact′s access right to the private data once the contact is removed from the social group. Zhang et al. [39] also propose a privacy-preserving verifiable profile matching scheme which is based on symmetric cryptosystem and thus improves efficiency. It relies on a pre-determined ordered set of attributes and uses it as a common secret shared by users. However, the scheme is not applicable to unordered sets of attributes such as random capabilities. Cristofaro and Tsudik [9] considered several flavors of private set intersection and constructed some provably secure protocols. They proposed efficient protocols for plain and authorized private set intersection and noted that, the choice between them depends on whether there is a need for client authorization and/ server unlinkability, as well as on servers ability to engage in pre-computation.

In matchmaking, persons make friends by match-pairing. A match-pair is made when two persons have some characteristics in common. In the quest to find the attributes two persons have in common, some protocols use either the trusted third party, the fully distributed technique or the hybrid technique.

With the use of the trusted third party, the trusted third party is involved in each step of the matchmaking process. The trusted third party collects personal attributes and location information, computes the intersection and notifies the matched-pair. Such protocol applications can be found in [12, 16, 17]. The use of the trusted third party has got some well-known problems.

The fully distributed technique requires no trusted third party in the whole matchmaking process. 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 users of this protocol are shared among multi-parties using Shamir secret sharing scheme, the computing of common attributes set are conducted among multi-parties as well [36]. The fully distributed technique can be found in matchmaking protocols in [6, 18, 22, 23].

The third technique in use is the hybrid technique − a combination of the two fore-mentioned techniques. In his technique, the trusted third party is needed only for the purpose of management and verification, and it does not participate in the matchmaking. In [8, 19, 20, 26, 27, 28, 29, 30, 35, 36] are matchmaking protocols based on the hybrid technique.

4 Our Matchmaking Protocol

In our quest to enable match-pair seekers find the most appropriate pair while at the same time prevent their private attributes from leaking, these protocols were formulated. In order for the users of this algorithm to achieve these said objectives, Algorithm 1 helps the initiator find a user who has enough attributes with him/her. In our protocol, the initiator sets a threshold number of attributes, $A_{Threshold}$ that a user should possess so as to qualify as a match-pair. Hence, a user of this protocol becomes a match-pair of the initiator if the number of attributes s/he has in common with the initiator is at least $A_{Threshold}$. The notations used in this paper are listed in Table 1.

The matchmaking protocol we are proposing comprise a certification authority (CA) that cannot be compromised and other users. These users consist an initiator, Alice and other persons called candidates. Each user has a portable device that has wireless interfaces such as Bluetooth or WiFi that is in communication range with each other. Among the m candidates, $k = 1, \ldots, m$ Alice wishes to find a candidate(s) who possesses attributes that are at least $A_{Threshold}$. The CA generates an RSA key-pair, (e_{CA}, d_{CA}) and $N = pq$, where p and q are large prime numbers. The CA makes N and e_{CA} public. Each person in the protocol also chooses a username and an ID, creates an RSA key-pair, (e, d). Alice creates an RSA key-pair (e_A, d_A) and each candidate also creates an RSA key-pair, (e_k, d_k). Alice makes e_A and her username public. Each candidate also makes e_k and username public.

The attributes of Alice are $A = \{a_1, a_2, \ldots, a_p\}$. Also, for all the $k = 1, \ldots, m$ candidates and $h = 1, \ldots, w$ attributes, the attributes of each candidate is $C_k =$

Table 1: Notations

Notation	Explanation
R_A	Random number chosen by Alice, $R_A \longleftarrow_r Z_{N/4}$
R_k	Random number chosen by each candidate, $R_k \longleftarrow_r Z_{N/4}$, $k = 1, \ldots, m$
R_B	Random number chosen by Bob, $R_A \longleftarrow_r Z_{N/4}$
ID_k	Identity of each candidate, $k = 1, \ldots, m$
ID_A	Identity of Alice
γ_{kh}	Computation to certify each candidates's attributes, $\gamma_{kh} = Sign_{d_{CA}}(ID_k \| C_{kh})$
α_j	Computation to certify Alice's attributes, $\alpha_j = Sign_{d_{CA}}(ID_j \| \alpha_j)$
$\mid I_{Ak} \mid$	Number of attributes Alice and each candidate have in common
$\mid I_{kA} \mid$	Number of attributes each candidate and Alice have in common

$\{c_{k1}, c_{k2}, \ldots, c_{kw}\}$.

Alice chooses a random number, $R_A \leftarrow_r Z_{N/4}$. Each candidate also chooses a random number $R_k \leftarrow_r Z_{N/4}$, $k = 1, \ldots, m$. In this matchmaking protocol, attributes are the same if they are semantically the same. Alice and each candidate then encrypt their attributes, ID, their random number, username, and the public key-pair of his/her RSA key using the public key of the CA. Each of the persons in the protocol sends his/her encrypted set to the CA.

Alice sends $E_{e_{CA}}\{A \parallel ID_A \parallel R_A \parallel username_A \parallel RSA_{publickey}, e_A\}$ to the CA. The CA certifies the attributes of Alice and the attributes become $A = \{(a_1, \alpha_1), (a_2, \alpha_2), \ldots, (a_p, \alpha_p)\}$, where $\alpha_j = sign_{d_{CA}}(ID_A \| a_j)$, $j = 1, \ldots, p$. The CA returns $A = \{(a_1, \alpha_1), (a_2, \alpha_2), \ldots, (a_p, \alpha_p)\}$ to Alice. Alice then exponentiates each of her attributes with her random number and sends to each candidate. Each candidate receives $\{a_1^{R_A}, a_2^{R_A}, \ldots, a_p^{R_A}\}$ from Alice.

Each candidate also sends $E_{e_{CA}}\{C_k \parallel ID_k \parallel R_k \parallel username_k \parallel RSA_{publickey}, e_k\}$ to the CA for certification. After certification of each candidate's attributes by the CA, the attributes become $C_k = \{(c_{k1}, \gamma_{k1}), (c_{k2}, \gamma_{k2}), \ldots, (c_{kw}, \gamma_{kw})\}$, where $\gamma_{kh} = sign_{d_{CA}}(ID_k \| c_{kh})$. Each candidate receives $C_k = \{(c_{k1}, \gamma_{k1}), (c_{k2}, \gamma_{k2}), \ldots, (c_{kw}, \gamma_{kw})\}$ from the CA. Each candidate exponentiates his/her attributes with the random number and sends to Alice. Hence, Alice receives $\{c_{k1}^{R_k}, c_{k2}^{R_k}, \ldots, c_{kw}^{R_k}\}$ from each candidate.

When Alice received $\{c_{k1}^{R_k}, c_{k2}^{R_k}, \ldots, c_{kw}^{R_k}\}$, she exponentiates it with her random number and returns $\{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\}$ to each candidate. Each candidate also exponentiates $\{a_1^{R_A}, a_2^{R_A}, \ldots, a_p^{R_A}\}$ with his/her random number and returns $\{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\}$ to Alice.

With the knowledge of $\{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\}$ and $\{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\}$, Alice computes the intersection between her and each candidate and outputs $|I_{Ak}| \in \{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\} \cap \{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\}$. Also, with the knowledge of $\{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\}$ and $\{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\}$, each candidate computes the intersection between him/her and Alice and outputs $|I_{kA}| \in \{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\} \cap \{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\}$. The intersections $|I_{Ak}|$ and $|I_{kA}|$ computed in Steps 11 and 12 of Algorithm 1 by Alice and each candidate respectively allow them to know the number of attributes they have in common with each other. The initiator, Alice then checks which candidate has attributes $|I_{Ak}| \geq A_{Threshold}$.

Algorithm 1 Computing the Number of Common Attributes

Require: The CA has an RSA key-pair, (e_{CA}, d_{CA}) makes N and e_{CA} public.

1: Alice creates an RSA key-pair (e_A, d_A) and chooses a random number $R_A \leftarrow_r Z_{N/4}$.
Also, each candidate creates an RSA key-pair (e_k, d_k) and chooses a random number $R_k \leftarrow_r Z_{N/4}$, for all $k = 1, \ldots, m$.
Alice and each candidate make their RSA-keys e_A and e_k public.

2: Alice has private attributes $A = \{a_1, a_2, \ldots, a_p\}$. Alice sends $E_{e_{CA}}\{A \| ID \| username_{Alice} \| e_A\}$ to the CA.

3: After certification by the CA, the attributes of Alice become $A = \{(a_1, \alpha_1), (a_2, \alpha_2), \ldots, (a_p, \alpha_p)\}$, where $\alpha_j = sign_{d_{CA}}(ID_A \| a_j)$.

4: Each of the k candidates has $C_k = \{c_{k1}, c_{k2}, \ldots, c_{kw}\}$ attributes, for all $k = 1, \ldots, m$ and $h = 1, \ldots, w$. Each candidate sends $E_{e_{CA}}\{C_k \| ID \| username_k \| e_k\}$ to the CA.

5: After certification by the CA, the attributes become $C_k = \{(c_{k1}, \gamma_{k1}), (c_{k2}, \gamma_{k2}), \ldots, (c_{kw}, \gamma_{kw})\}$, where $\gamma_{kh} = sign_{d_{CA}}(ID_k \| c_{kh})$.

6: Alice exponentiates each of her attributes with her random number and sends $\{a_1^{R_A}, a_2^{R_A}, \ldots, a_p^{R_A}\}$ to each candidate.

7: For all $k = 1, \ldots, m$ and $h = 1, \ldots, w$, each candidate exponentiates his/her attributes with the random number and sends $\{c_{k1}^{R_k}, c_{k2}^{R_k}, \ldots, c_{kw}^{R_k}\}$ to Alice.

8: Alice computes $\{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\}$ and sends to each candidate.

9: Each candidate also computes and sends $\{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\}$ to Alice.

10: Alice computes the intersection between her and each candidate and outputs $|I_{Ak}| \in \{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\} \cap \{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\}$.

11: Each candidate computes the intersection between him/her and Alice and outputs $|I_{kA}| \in \{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\} \cap \{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\}$.

12: The intersections $|I_{Ak}|$ and $|I_{kA}|$ computed in Steps 11 and 12 by Alice and each candidate respectively allows each of them to know the number of attributes each has in common with the other.

Alice sends her username, the intersection she computed and the username of the candidate whose attributes is at least $A_{Threshold}$ to the CA. Alice sends

$E_{e_{CA}}\{username_{Alice}|||I_{Ak}|||username_k\}$ to the CA. Each candidate also sends his/her username, the intersection computed and the username of Alice to the CA. Thus each candidate sends $E_{e_{CA}}\{username_k|||I_{kA}|||username_{Alice}\}$ to the CA. The CA verifies $|I_{Ak}|$ and $|I_{kA}|$. If $|I_{Ak}| = |I_{kA}|$, the CA notifies Alice and the candidate(s) of a successful match. However, if $|I_{Ak}| \neq |I_{kA}|$ the CA checks who cheated. The CA the removes the cheat from the protocol users.

For simplicity, lets us assume Bob was the only candidate whose attributes was at least $A_{Threshold}$. Let R_B be the random number of Bob. Alice and Bob exchange their random numbers using Algorithm 2. Algorithm 2 is an authenticated Diffie-Hellman protocol. Alice and Bob agree on a primitive prime number, g. At the end of Algorithm 2, both Alice and Bob will know each other's random number. Alice receives Bob's random number, R_B and Bob also receives Alice's random number, R_A. At the end of Algorithm 2, Alice sends the random number received from Bob to the CA. Thus Alice sends $E_{e_{CA}}\{username_{Alice}||username_{Bob}||R_B\}$ to the CA. Bob also sends the random number he received from Alice to the CA by sending $E_{e_{CA}}\{username_{Bob}||username_{Alice}||R_A\}$. The CA verifies if the random number Alice sent to Bob is the same as that in Step 1 of Algorithm 1. Also, the CA verifies if the random number Bob sent to Alice is the same as that in Step 1 of Algorithm 1. If the CA observes that the random numbers are the same, the CA then notifies them. Hence, with the knowledge of R_A and R_B both Alice and Bob can be able to know the actual attributes they have in common.

Algorithm 2 Authenticated Diffie-Hellman protocol for exchanging the random numbers of the matched-pair

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

1: Using the generator g, Alice computes and sends $g^{R_A} = Enc(g^{R_A} \parallel ID_A)$ to Bob.
2: Bob using the generator, g, computes $g^{R_B} = Enc(g^{R_B} \parallel ID_B)$ and sends $g^{R_A} \parallel g^{R_B} \parallel Sign_{Bob}(g^{R_A} \parallel g^{R_B} \parallel ID_A)$ to Alice.
3: Alice computes and sends $Sign_{Alice}(g^{R_A} \parallel g^{R_B} \parallel ID_A)$ to Bob.
4: Alice computes $(g^{R_A})^{R_B}$ and Bob also computes $(g^{R_B})^{R_A}$

4.1 Experimental Implementation

Our protocol for computing the number of common attributes was simulated in java. We focused only on the execution time without considering the communication time. In this simulation, the execution time is mainly decided by the number of participants and the number of attributes they possess. The prime numbers p and q we chosen to be 1024 bits with RSA modulus of 1024 bits. Also, each attribute was represented by 64 bits. The execution time for the protocol was measured. The time duration to run the protocol between an initiator and a candidate constituted the execution time. The number of attributes of the initiator was kept constant whilst the number of attributes of the candidates were varied. In the experiment, the number of users was varied $k = 1, 5, 10, 15, 20, 25$. The initiator has the same number of attributes but the number of attributes of each candidate varied $h = 5, 10, 15, 20$. The protocol was simulated on an hp-compaq laptop with 2.10 GHz processor and 4G RAM. In order to ensure the accuracy of the execution time, the average of 80 repeated execution times was used. Figure 1 is the graph of the execution times for our protocol for the users and their attributes. The x-axis shows the number of users and the y-axis shows the execution time. The graph shows the execution times for the varying number of users and users attributes. It can be observed that the execution times increases as the number of users and attributes increases.

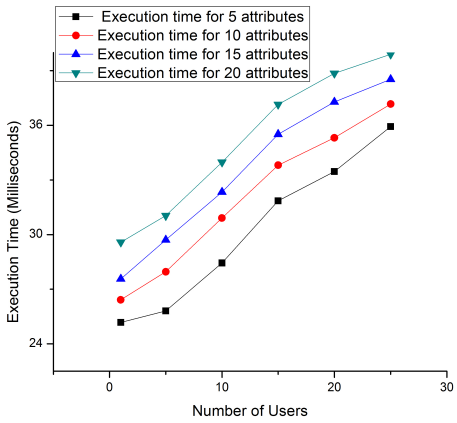

Figure 1: Comparison of execution time for the number of attributes

4.2 Security Analysis

In Algorithm 1, CA certifies users' private attributes to be used in the protocol. The certification of the attributes binds the attributes to the attribute owners. Hence, a user(s) cannot modify the attributes so as to gain more information from others in the protocol. Also, in order to prevent the attribute owners from modifying their attributes after the certification, the CA does the following computation. The CA computes and sends $A = \{(a_1, \alpha_1), (a_2, \alpha_2), \ldots, (a_p, \alpha_p)\}$ and $C_k = \{(c_{k1}, \alpha_{k1}), (c_{k2}, \alpha_{k2}), \ldots, (c_{kw}, \alpha_{kw})\}$ to Alice and each of the other candidates respectively.

In Step 6 of Algorithm 1, in order to prevent a candidate from knowing her attributes, Alice exponentiates her attributes with her random number. This exponentiation will prevent a candidate from being able to know the attribute a_i from $a_i^{R_A}$, $i = 1, \ldots, p$ in polynomial time. Each candidate also in Step 7 of Algorithm 1 exponentiates each of the attributes so as to prevent Alice from being able to know c_{kh} from $c_{kh}^{R_k}$, $k = 1, \ldots, m$ and $h = 1, \ldots, w$ in polynomial time.

In Step 11 of Algorithm 1, Alice computes the intersection set between her and each candidate. The computation of the intersection, $|I_{Ak}|$ ϵ

$\{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\} \cap \{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\}$ allows Alice know only the number of attributes she has in common with each candidate. Likewise, the computation of the intersection, $|I_{kA}| \in \{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\} \cap \{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\}$ by each candidate allows him/her know only the number of attributes the candidate has in common with Alice. At the end of Algorithm 1, the initiator as well as each candidate will know the number of attributes they have in common with each other. Alice then checks which candidate's $|I_{Ak}| \geq A_{Threshold}$. The candidate with $|I_{Ak}| \geq A_{Threshold}$ then becomes her match-pair. In this protocol, only candidates with $|I_{Ak}| \geq A_{Threshold}$ proceed to Algorithm 2. If a person decides to terminate the protocol because the number of attributes they have in common is small, the actual attributes will be preserved.

In order to prevent semi-honest attack on Algorithm 1, Alice and the candidate(s) who has $|I_{Ak}| \geq A_{Threshold}$ attributes send their intersection to the CA. The CA then verifies if $|I_{Ak}| = |I_{kA}|$. If $|I_{Ak}| = |I_{kA}|$ is verified successfully, the protocol continues to Algorithm 2. However if $|I_{Ak}| \neq |I_{kA}|$, the protocol is terminated and the CA checks who has cheated and removes the cheat from the list of the protocol users.

In order to exchange their random numbers securely, Alice and Bob execute Algorithm 2. The authenticated Diffie-Hellman protocol in Algorithm 2 ensures that, there is no meet-in-the middle attack by a malicious persons. To further ensure that the protocol for this matchmaking is secured, Alice and Bob send the random number they received from each other to the CA. The CA then verifies if the correct random numbers have been exchanged. This is done so as to prevent semi-honest attack on the protocol. When the CA observes that the random numbers are not the same, the protocol is terminated. The CA then checks who might have cheated and remove the cheat from the list of the protocol users.

4.2.1 Correctness of the Protocol

In Step 8 of Algorithm 1, Alice sends $\{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \cdots, c_{kw}^{R_k R_A}\}$ to each candidate. Likewise, each candidate sends $\{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\}$ to Alice in step 9. Alice computes and outputs the intersection $|I_{Ak}| \in \{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\} \cap \{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\}$. Each candidate also computes and outputs the intersection $|I_{kA}| \in \{c_{k1}^{R_k R_A}, c_{k2}^{R_k R_A}, \ldots, c_{kw}^{R_k R_A}\} \cap \{a_1^{R_A R_k}, a_2^{R_A R_k}, \ldots, a_p^{R_A R_k}\}$. The number of attributes in $|I_{Ak}|$ and $|I_{kA}|$ are the same, ($|I_{Ak}| = |I_{kA}|$). Hence, Algorithm 1 is correct. Also, at the end of Algorithm 2, Alice computes $(g^{R_A})^{R_B}$; Bob also computes $(g^{R_B})^{R_A}$. Since $(g^{R_A})^{R_B}$ and $(g^{R_B})^{R_A}$ are the same, Algorithm 2 is also correct.

4.2.2 Preventing Other Attacks on the Protocol

In this paper, attack by persons outside the protocol is not possible as a person needs to register with the CA to be able to run the protocol. Also, part from Alice who knows the number of candidates that are running the protocol with her, no other protocol user does. As the candidates do not know about the other candidates in the protocol, they cannot collude to know all of Alice's attributes. Hence, our protocol is collusion resistant. Nonspoofability of the other users' attributes is another characteristic of our protocol. The attributes of the users in the protocol are certified hence, a user cannot query another$'s$ attributes without his/her knowledge.

5 Discussion, Implication, and Conclusion

in real life, people who have many characteristics in common tend to be good friends. This behavior is also used on social networks. Hence on social networks, the ability to know the number of attributes a person has in common with the other before they become friends is also very important.

Knowing the number of attributes a person has in common with the other before they exchange their attributes prevents the termination of the protocol; keeps the privacy and security of the attributes. Hence, by implication helps users feel more confident in using such protocols.

Matchmaking has becoming very popular on mobile social networks. Hence, there is the need for secure and privacy-preserving matchmaking protocol for MSN. This research paper has effective proposed a matchmaking protocol that will enable an individual find a match on MSN.

The quest to know the number of attributes two individuals, each having a private set of attributes have in common is becoming very important in matchmaking. This knowledge can be extended to any database to ascertain the items they have in common. Cloud computing is gaining more popularity as a data storage facility hence it is recommended the application of the knowledge from this paper to compare the content of clouds.

References

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

[2] G. Ateniese, E. De Cristofaro, and G. Tsudik, "(if) size matters: Size-hiding private set intersection," in *14th International Conference on Practice and Theory in Public Key Cryptography Conference on Public Key Cryptography (PKC'11)*, pp. 156–173, Springer-Verlag, 2011.

[3] R. W. Baldwin and W. Gramlich, "Cryptographic protocol for trustable match making," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 92–100, 1985.

[4] 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 *Proceedings of 12th International Conference on Practice and Theory in Public Key Cryptography (PKC'09)*, LNCS 5445, pp. 196–214, Springer, 2009.

[5] J. Camenisch and G. M. Zaverucha, "Private intersection of certified sets," in *Financial Cryptography and Data Security*, LNCS 5628, pp. 108–127, Springer, 2009.

[6] A. C. Champion, Z. Yang, B. Zhang, J. Dai, D. Xuan, and Du Li, "E-smalltalker: A distributed mobile system for social networking in physical proximity," *IEEE Transactions on Parallel Distribution Systems*, vol. 24, no. 8, pp. 1535–1545, 2013.

[7] M. Yu Chen, C. C. Yang, and M. S. Hwang, "Privacy protection data access control," *International Journal of Network Security*, vol. 15, no. 6, pp. 411–419, 2013.

[8] E. De Cristofaro, A. Durussel, and I. Aad, "Reclaiming privacy for smartphone applications," in *Proceedings of Ninth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom'11)*, pp. 84–92, Seattle, USA, 2011.

[9] E. De Cristofaro, Y. Lu, and G. Tsudik, "Efficient techniques for privacy-preserving sharing of sensitive information," in *Proceedings of the 4th International Conference on Trust and Trustworthy Computing (TRUST'11)*, pp. 239–253, Springer-Verlag, 2011.

[10] E. De Cristofaro and G. Tsudik, "Practical private set intersection protocols with linear complexity," in *Proceedings of 14th International Conference on Financial Cryptography and Data Security*, pp. 143–159, 2010.

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

[12] N. Eagle and A. Pentland, "Social serendipity: Mobilizing social software," *IEEE Pervasive Computing*, vol. 4, no. 2, pp. 28–34, 2005.

[13] M. J. Freedman, K. Nissim, and B. Pinkas, "Efficient private matching and set intersection," in *Advances in Cryptology (EUROCRYPT'04)*, LNCS 2267, pp. 1–9, Springer, 2004.

[14] C. Hazay and Y. Lindell, "Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries," in *Proceedings of 5th Conference on Theory of Cryptography (TCC'08)*, pp. 155–175, 2008.

[15] L. Kissner and D. X. Song, "Privacy-preserving set operations," in *Advances in Cryptology (CRYPTO'05)*, pp. 241–257, 2005.

[16] J. Kjeldskov and J. Paay, "Just-for-us: A context-aware mobile information system facilitating sociality," in *Proceedings of 7th ACM International Conference on Human Computer Interaction with Mobile Devices & Services (MobileHCI'05)*, pp. 23–30, New York, USA, 2005.

[17] K. A. Li, T. Y. Sohn, S. Huang, and W. G. Griswold, "Peopletones: A system for the detection and notification of buddy proximity on mobile phones," in *Proceedings of 6th ACM International Conference on Mobile Systems, Applications, and Services (MobiSys'08)*, pp. 160–173, New York, USA, 2008.

[18] M. Li, N. Cao, S. Yu, and W. Lou, "Findu: Privacy-preserving personal profile matching in mobile social networks," in *Proceedings of 30th IEEE International Conference on Computer Communications*, pp. 2435–2443, Shanghai, China, 2011.

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

[20] X. Liang, R. Lu, X. Lin, and X. Shen, *Security and Privacy in Mobile Social Networks*, Springer Briefs in Computer Science, Springer Berlin Heidelberg: Springer, 2013.

[21] H. Lin, S. M. Chow, D. Xing, Y. Fang, and Z. Cao, "Privacy-preserving friend search over online social networks," *IACR Cryptology ePrint Archive*, pp. 445, 2011.

[22] R. Lu, X. Lin, X. Liang, and X. Shen, "A secure handshake scheme with symptoms-matching for mhealthcare social network," *Mobile Network Applications*, vol. 16, no. 6, pp. 683–694, 2011.

[23] R. Lu, X. Lin, and X. Shen, "SPOC: A secure and privacy-preserving opportunistic computing framework for mobile-healthcare emergency," *IEEE Transactions on Parallel Distribution Systems*, vol. 24, no. 3, pp. 614–624, 2013.

[24] C. Meadows, "A more efficient cryptographic matchmaking protocol for use in the absence of a continuously available third party," in *Proceedings of IEEE Symposium on Security and Privacy*, pp. 134–137, 1986.

[25] Y. Sang and H. Shen, "Privacy preserving set intersection protocol secure against malicious behaviors," in *Proceedings of Eighth IEEE International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'07)*, pp. 461–468, 2007.

[26] S. Sarpong and C. Xu, "Efficient privacy-preserving attribute matchmaking protocol for proximity-based mobile social networks," in *Proceedings of A10th International Conference on dvanced Data Mining and Applications (ADMA'14)*, LNCS 8933, pp. 305–318, Springer-Verlag, 2014.

[27] S. Sarpong and C. Xu, "Provably secure attribute matchmaking protocol for mobile social network secure against malicious users," in *1st International Conference on Computer, Network Security and Communication Engineering (CNSCE'14)*, pp. 362–366, Shenzhen, China, 2014.

[28] S. Sarpong and C. Xu, "A collusion-resistant privacy-preserving attribute matchmaking for mobile social

networks," *International Journal of Innovative Science, Engineering and Technology*, vol. 2, pp. 485–495, 2015.

[29] S. Sarpong and C. Xu, "Privacy-preserving attribute matchmaking for proximity-based mobile social networks," *International Journal of Security and its Applications*, vol. 9, no. 5, pp. 217–230, 2015.

[30] S. Sarpong, C. Xu, and X. Zhang, "An authenticated privacy-preserving attribute matchmaking protocol for mobile social networks," *International Journal of Network Security*, vol. 17, no. 3, pp. 357–364, 2015.

[31] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proceedings of Advances in Cryptology (CRYPTO'85)*, pp. 47–53, Springer-Verlag, 1985.

[32] Ji S. Shin and V. D. Gligor, "A new privacy-enhanced matchmaking protocol," *IEICE Transactions on Communications*, vol. E96-B, no. 8, pp. 2049–2059, 2013.

[33] J. Sun, X. Zhu, and Y. Fang, "A privacy-preserving scheme for online social networks with efficient revocation," in *Proceedings 29th IEEE Conference on Information Communications (INFOCOM'10)*, pp. 2516–2524, 2010.

[34] J. Vaidya and C. Clifton, "Secure set intersection cardinality with application to association rule mining," *Journal of Computer Security*, vol. 13, no. 4, pp. 593–622, 2005.

[35] Y. Wang, J. Hou, Y. W. Tan, and X. Nie, "A recommendation-based matchmaking scheme for multiple mobile social networks against private data leakage," in *Proceedings of 1st International Conference on Information Technology and Quantitative Management (ITQM'13)*, pp. 781–788, 2013.

[36] 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 *1Proceedings of 1th IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TRUSTCOM'12)*, pp. 609–615, 2012.

[37] Qi Xie and U. Hengartner, "Privacy-preserving matchmaking for mobile social networking secure against malicious users," in *Proceedings of Ninth IEEE Annual Conference on Privacy, Security and Trust (PST'11)*, pp. 252–259, 2011.

[38] Q. Ye, H. Wang, and J. Pieprzyk, "Distributed private matching and set operations," in *Proceedings of the 4th International Conference on Information Security Practice and Experience (ISPEC'08)*, pp. 347–360, 2008.

[39] C. Zhang, J. Sun, X. Zhu, and Y. Fang, "Privacy and security for online social networks: challenges and opportunities," *IEEE Network*, vol. 24, no. 4, pp. 13–18, 2010.

[40] K. Zhang and R. Needham, "A private matchmaking protocol," 2001. (`http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.54.835&rep=rep1&type=pdf`)

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).

Xiaojun ZHANG, received his B.Sc. degree in mathematics and applied mathematics at Hebei Normal University in 2009. He also received his M.Sc. degree in pure mathematics at Guangxi University, P. R. China, in 2012. He is currently pursuing his Ph.D degree in Information Security at University of Electronic Science and Technology of China (UESTC). He is currently engaged in Cryptography, Network Security and Cloud Computing Security.