A Secure and Efficient Privacy-Preserving Attribute Matchmaking Protocol for Mobile Social Networks

K. Arthi[1], M. Chandramouli Reddy[2]
(Corresponding author: M. Chandramouli Reddy)
Department of Information Technology, Veltech Technical University[1]
Department of Computer Science & Engineering, Veltech Technical University[2]
42, Avadi-Vel Tech Road, Avadi, Chennai, Tamil Nadu 600062, India
(Email: mouli.veltech@gmail.com)
(Received Feb. 24, 2016; revised and accepted Apr. 25 & May 7, 2016)

Abstract

The advances in mobile and communication technologies lead to advancement of Mobile Social Networks (MSNs). MSN changed the way people communicate and exchange the private and sensitive information among the friend groups via mobile phones. Due to the involvement of private and sensitive information, MSN demands for efficient and privacy-preserving matchmaking protocols to prevent the unintended data (attribute) leakage. Many existing matchmaking protocols are based on user's private and specific data. Malicious participants may opt their attribute set arbitrarily so as to discover more information about the attributes of an honest participant. Hence, there is great chance of information leakage to a dishonest participant. In this context, Sarpong et al. had proposed a first of its kind of an authenticated hybrid matchmaking protocol that will help match-pair initiators to find an appropriate pair which satisfies the pre-defined threshold number of common attributes. Sarpong et al. had claimed that their protocol restricts attribute leakage to unintended participants and proved to be secure. Unfortunately, in Sarpong et al. scheme, after thorough analysis, we demonstrate that, their scheme suffers from data (attribute) leakage, in which the initiator and the participant can compute or achieve all the attributes of each other. Also we show that Sarpong et al. scheme requires huge computation and communication cost. As a part of our contribution we will propose an efficient and secure match making protocol which is light weight and restricts attribute leakage to the participants.

Keywords: Matchmaking protocols, mobile social networks, privacy-preserving attribute matchmaking protocol

1 Introduction

The advances in mobile and communication technologies lead to advancement of traditional online social network to Mobile Social Networks (MSNs). MSN facilitates real time personal and user specific data sharing and instant messaging among friend groups. Due to the exchange of private and shared information among the participants, finding a matching pair privately is a critical requirement in MSN.

A private matchmaking is a primary feature of private set intersection. Matchmaking protocol is a critical requirement for MSN in which two or more mutually mistrustful parties A and B consists of attribute sets SA and SB desire to compute together the intersection in such a way that both A and B should not take any information particular to the other opponent. A and B must learn only the common attributes among them i.e. SA∩SB nothing more.

In literature, many match making algorithms has been proposed based on various parameters. Few matchmaking protocols [1, 6, 7] has been proposed based on Certificate Authority C.A, in which C.A authenticates the entities attributes. Another matchmaking technique is fully distributed [9], which eliminates C.A. The participants perform the distribution of attributes among themselves, computing the intersection set. The initiator and the multi parties exchange their attributes using Shamir secret sharing scheme [3]. The Hybrid technique [12] is a commonly used technique in which the CA performs only the verification of attributes and managing the communication among the entities. The protocol participants will perform the attribute sharing and matchmaking opera-

tions. Recently Huang et al. [14] had proposed an Identity Based Encryption scheme for match making in social networks.

However, in this context in 2015, Chiou et al. [2] and Sarpong et al. [15] had proposed matchmaking protocols in which the initiator finds the best match among multiple participants who has the maximum similar attribute as the initiator. Sarpong et al. claimed that their scheme protects user's attributes from unnecessary leakage to unintended persons. In this manuscript after thorough analysis of Sarpong et al. scheme, we will demonstrate that in Sarpong et al. scheme, the participants can achieve the attributes of other participants and requires huge computation and communication cost.

As a part of our contribution, we will propose a secure and light weight matchmaking protocol for MSN, which resists the pitfalls in Sarpong and other related schemes.

The remaining of the paper is systematized as follows: In Section 2, we will give a brief review on system architecture. In Section 3, we will briefly discuss on Sarpong et al. [15] scheme. In Section 4, we discuss on the security pitfalls in Sarpong et al. scheme. In Section 5, the anomalies in Sarpong et al. scheme are discussed. Our proposed matchmaking protocol is presented in Section 6. In Section 7 we deliberate on informal security analysis of our proposed scheme. In Section 8, we deliberate on formal security analysis of our proposed scheme using widely accepted random oracle model. We discuss on Simulating experiments and performance evaluations are provided in Section 9.

2 System Architecture and Design Goals

Our system architecture consists of mainly three entities:a user (initiator) to find the best match among multiple participants (called participants) and a trusted certificate authority (CA), as depicted in Figure 1.

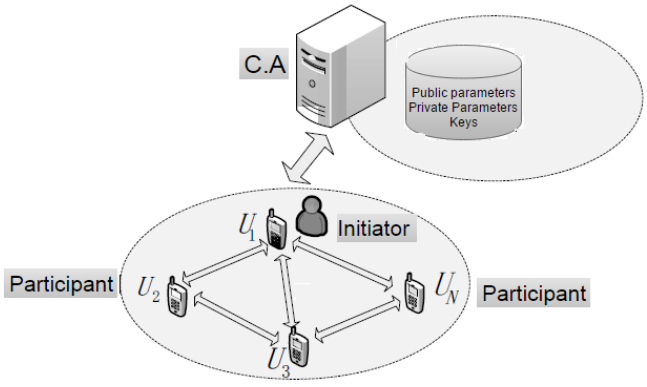

Figure 1: The system architecture

We will follow below mentioned privacy levels similar to [5].

Privacy Level 1: On completion of execution of matchmaking protocol,the initiator and each potential friend (participant)must identify only the intersection set and its size.

Privacy Level 2: On completion of execution of matchmaking protocol,the initiator and each potential friend (participant) must know only the ranking of the size of the intersection set mutually. Apart from these, no other information should be intercepted by the participants.

3 Brief Review of Sarpong et al. Matchmaking Algorithm

Assume Alice is the initiator of the protocol to find out the closest match among 'm' participant's (for brevity, we assume that Alice is communicating with a single participant Bob to find out the common attributes. The other participants also perform and exchange similar messages as Bob with Alice. Alice also exchanges same messages as it exchanges with Bob.) having portable devices and can connect with each other using PAN or Bluetooth or Wifi. $A_{Threshold}$ is the threshold value for the attribute matching set by the initiator Alice, i.e. to qualify as a match pair for initiator, there should be minimum of $A_{Threshold}$ number of common attributes between pairs. The initiator Alice consists of 'm' attributes, i.e. $a = \{a_1, a_2, \cdots, a_m\}$ and Bob consists 'p' attributes, i.e. $b = \{b_1, b_2, \cdots, b_p\}$. In the matchmaking, if two attributes are semantically same, then only they are treated as the same.

3.1 Key Generation

K1. Alice and Bob computes RSA key pairs (e_A, d_A), (e_B, d_B) respectively using p, q which are large prime numbers, where e_A, e_B are the public variables.

K2. CA computes RSA key pair is (e, d), where N=p*q.

K3. CA makes $< e, N >$ public.

3.2 Attribute Certification

A1. The attributes of Alice and Bob are $a = \{a_1, a_2, \cdots, a_m\}$ and $b = \{b_1, b_2, \cdots, b_k\}$.

A2. Alice exponentiates her attribute set using the public key of CA, i.e. 'e'. $a^e = \{a_1{}^e, a_2{}^e, \cdots, a_m{}^e\}$.

A3. Bob also exponentiates his attributes as $b^e = \{b_1{}^e, b_2{}^e, \cdots, b_k{}^e\}$.

A4. Alice to get the attributes certified by CA, forwards a message $E_e\{a^e||ID_A||UN_A||e, e_A\}$ to CA which contains the attribute set computed in A2, its identity, user name, its public key and CA public key. The message is encrypted with the CA public key, i.e. 'e'.

A5. Bob also to get the attributes certified by CA, forwards a message $E_e\{b^e||ID_B||UN_B||e||e_B\}$ to CA which contains the attribute set computed in A3, its identity, user name, its public key and CA public key. The message is encrypted with the CA public key, i.e. 'e'.

A6. The CA certifies the Alice attributes and returns $A = \{(a_1, s_1), (a_2, s_2), \cdots, (a_m, s_m)\}$ to Alice, where $s_i = H(ID_A||a_i)^d \bmod N$ using its private key 'd'.

A7. The CA also certifies the Bob attributes and returns $B = \{(b_1, \sigma_1), (b_2, \sigma_2), \cdots, (b_k, \sigma_k)\}$ to Bob, where $\sigma_1 = H(ID_B||b_1) \bmod N$.

3.3 Matchmaking Phase

M1. On getting the attributes certified by the CA, the private attributes of Alice and Bob becomes $A = \{(a_1, s_1), (a_2, s_2), \cdots, (a_m, s_m)\}$, $B = \{(b_1, \sigma_1), (b_2, \sigma_2), \cdots, (b_k, \sigma_k)\}$ respectively.

Challenge Phase:

M2. Alice picks 'm' arbitrary random numbers R_i for each attribute $i = \{1, 2, \cdots, m\}$ and computes $MA_i = S_i.g^{R_i} \bmod N$, i.e. $MA_1 = s_1.g^{R_1} \bmod N$, $MA_2 = s_2.g^{R_2} \bmod N$, $MA_3 = s_3.g^{R_3} \bmod N$ and sends $MES_1 = \{MA_1, MA_2, \cdots, MA_m\}$ to Bob.

M3. Bob also chooses an arbitrary numbers P_k for each attribute $k = \{1, 2, \cdots, k\}$ and computes $MB_k = \sigma_k.g^{P_k} \bmod N$, i.e. $MB_1 = \sigma_1.g^{P_1} = H(ID_B||b_1).g^{P_1} \bmod N$, $MB_2 = \sigma_2.g^{P_2} \bmod N = H(ID_B||b_2).g^{P_2} \bmod N$, and sends $MES_2 = \{MB_1, MB_2, \cdots, MB_k\}$ to Alice.

3.4 Encoding Phase

M4. Alice chooses an arbitrary number R_a and computes $Z_A = g^{e.R_a} \bmod N$, $MB_k^* = (MB_k)^{e*R_a} = \{{MB_1}^{e.R_a}, {MB_2}^{e.R_a}, \cdots, {MB_m}^{e.R_a}\} = \{(H(ID_B||b_1).g^{P_1}).^{e.R_a}, (H(ID_B||b_2).g^{P_2}).^{e.R_a}, \cdots, (H(ID_B||b_m).g^{P_m}).^{e.R_a}.\}$.

M5. Alice performs arbitrary permutation $RPA = \zeta\{a_1, a_2, \cdots, a_m\}^{R_a} = \zeta\{{a_1}^{R_a}, {a_2}^{R_a}, \cdots, {a_m}^{R_a}\}$ and sends $MES_3 = \{Z_A||MB_k^*||RPA\}$ to Bob.

M6. Bob also opts an arbitrary number R_b and computes $ZB = g^{e.R_b} \bmod N$, $(MES_1)^{e.R_b} = \{M1^{e.R_b}, M_2^{e.R_b}, M_3^{e.R_b}, \cdots, M_k^{e.R_b}\} = \{(s_1.g^{R_1})^{e.R_b}, (s_2.g^{R_2})^{e.R_b}, \cdots, (S_k.g^{R_m})^{e.R_b}.\}$.

M7. Bob chooses an arbitrary permutation $RPB = \zeta\{b_1^{R_b}, b_2^{R_b}, \cdots, b_k^{R_b}\}$ and sends $MES_4 = \{Z_k||(MES_1)^{e.R_b}||RPB\}$ to Alice.

3.5 Set Intersection Phase

M8. Alice sends her signed message $Sig_{d_A}(ID_A \,||\, MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$ to Bob.

M9. Bob also sends his signed message $Sig_{d_B}(ID_B \,||\, MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$ to Alice.

M10. Now Alice and Bob verify that received MES_1, MES_2, MES_3, MES_4 values are equivalent to the received or computed values in the previous steps.

M11. Alice share her random number to Bob by sending $Sig_{d_A}(ID_A||ID_B||R_a)$. Similarly Bob also shares his arbitrary number by sending $Sig_{d_B}(ID_B||ID_A||R_b)$.

3.6 Recovery Phase

M12. Alice computes a list $KA = \zeta A\{{a_1}^{R_aR_b}, {a_2}^{R_aR_b}, \cdots, {a_m}^{R_aR_b}\}$ and direct to Bob. Bob also computes $KB = \zeta B\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}$ and send it to Alice.

M13. In order to know the actual common attributes, Alice sends her random permutation by encrypting with the Bob public key, i.e. e_B, E_{e_B} (ζA). Similarly, Bob sends his random permutations to Alice by encrypting with the Alice public key, i.e. e_a, $E_{e_A}(\zeta B)$.

M14. Alice already knowing ζB, can able to compute ζB^{-1} and retrieves $\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}$, similarly Bob able to compute ζA^{-1} and recover $\{{a_1}^{R_aR_b}, {a_2}^{R_aR_b}, \cdots, {a_m}^{R_aR_b}\}$. Now both Alice and Bob know their actual common attributes.

4 Cryptanalysis of Sarpong et al. Algorithm

In this section we do a thoughtful security analysis of Sarpong et al. [15] scheme. Based on the actions perform by the attackers,to intercept the information exchanged among the protocol entities,the attackers in the system are classified into two types i.e.malicious and semi-honest.The malicious or active attackers deviate the protocol, and try to achieve the private information from the protocol participants by providing the forged attributes. The semi-honest or passive attackers are intrusive, follows the protocol rules as specified and try to achieve extra information from the messages exchanged in the protocol execution.

4.1 Failure to Resist Malicious Attack

In Sarpong et al. [15] scheme, in M13 of matching phase, Alice sends its random permutation, i.e. $E_{e_B}(\zeta A)$ by encrypting with the Bob public key. Similarly Bob sends its random permutation $E_{e_A}(\zeta B)$ by encrypting with the Alice public key. On receiving the encrypted message

$E_{e_A}(\zeta B)$, Alice perform following steps as depicted below:

Step 1: Decrypts $E_{e_A}(\zeta B)$ using its private key d_A, i.e. $D_{d_A}E_{e_A}(\zeta B) = \zeta B$.

Step 2: Alice performs inverse operation ζB^{-1} on KB, i.e. $\zeta B^{-1}(KB) = \zeta B^{-1}\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}$ to retrieve original list, i.e. $\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}$.

Step 3: In M11 of matching phase Bob sends the message $Sig_{d_B}(ID_B||ID_A||R_b)$ to Alice. Alice retrieves $\{ID_B, ID_A, R_b\}$ from the received message. Alice already knows her R_a, hence Alice can perform an inverse operation on each received value in $\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}$, i.e. $\{{b_1}^{R_bR_a}, {b_2}^{R_bR_a}, \cdots, {b_k}^{R_bR_a}\}{R_b}^{-1}{R_a}^{-1} = \{b_1, b_2, b_3, b_4, \cdots, b_k\}$. Hence, Alice comes to know all the attributes of Bob, along with the common attributes. Similar is the case with Bob, in which Bob also comes to know all the attributes of Alice along with the common attributes by executing the above steps similar to Bob. Therefore we can conclude that, Sarpong et al. scheme fails to achieve the primary requirement of match making algorithm, in which the participant and initiator must know only the common attributes.

5 Pitfalls or Anomalies in Sarpong et al. Algorithm

5.1 Requires Huge Communication Cost

In M2 and M5 steps of match making process, Alice sends MES_1 and MES_3 to Bob respectively. In M8, Alice again forwards MES_1, MES_3 to Bob in a message $Sig_{d_A}(ID_A||MES_1||MES_2||MES_3||MES_4)$. Bob, on receiving the message $Sig_{d_A}(ID_A \;||\; MES_1 \;||\; MES_2 \;||\; MES_3 \;||\; MES_4)$, decrypts the message to get $\{ID_A, MES_1, MES_2, MES_3, MES_4\}$ and uses $MES_1, MES_2, MES_3, MES_4$ to validate, whether the transferred and received values are valid or not. To validate the messages transferred, a message digest operations like hash functions Eg: SHA-1 etc can be used, which outputs a fixed length data, hence reduces the necessitate to transfer full messages.

Similar is the case with the Bob. In M3 Bob sends MES_2, in M7 Bob sends MES_4 to Alice. In M9 Bob again sends these messages in the form of $Sig_{d_B}(ID_B||MES_1||MES_2||MES_3||MES_4)$ to Alice, which consumes huge communication cost.

5.2 Requires Huge Computation Cost

In M2 and M3 steps of match making process, Alice and Bob selects 'm' and 'p' arbitrary numbers respectively. Alice computes $MA_i = s_i.gR_i \bmod N$, where $1 <= i <= m$. Similarly Bob computes $MB_k = \sigma_k.gP_k = H(ID_B||b_k).gP_k \bmod N$, where $i <= k <= p$. Totally for one participant and one initiator, the Sarpong et al. schemes $k*m$ random numbers, which requires huge computation cost Alice.

6 Our Proposed Scheme

In this section we present our improved scheme over Sarpong et al. [15] scheme. The Key Generation and Attribute Certification phases of our proposed scheme are similar to Sarpong et al. scheme. We will start from matchmaking phase.Even though the protocol runs between Alice and 'm' participant, for brevity we consider only Bob as another participant of the protocol.

The main contributions of our work are:

1) An enhanced matchmaking protocol for MSN is proposed, which is based on the trusted certification authority (TCA) and provides a better privacy preserving by introducing the protocol's privacy levels.

2) The theoretical analysis is performed to prove the correctness and security of the protocol. Simulating experiments are conducted to evaluate the efficiency of the protocol.

3) We discuss the arbitration mechanisms for the protocol to detect malicious users who are cheating the others.

6.1 Matchmaking Phase

M1. On getting the attributes certified by the CA, the private attributes of Alice and Bob becomes $A = \{(a_1, s_1), (a_2, s_2), \cdots, (a_m, s_m)\}$, $B = \{(b_1, \sigma_1), (b_2, \sigma_2), \cdots, (b_k, \sigma_k)\}$, respectively.

Challenge Phase:

M2. Alice picks a single arbitrary random number R_1, and computes $MA_1 = S_i.g^{R_1} \bmod N$, i.e. $MA_1 = s_1.g^{R_1} \bmod N$, $MA_2 = s_2.g^{R_2} \bmod N$, $MA3 = s3.g^{R_3} \bmod N$ and sends $MES_1 = \{MA_1, MA_2, \cdots, MA_m\}$ to Bob.

M3. Each participant also chooses an arbitrary numbers P_1 computes $MB_k = \sigma_k.g^{P_1} \bmod N$, i.e. $MB_1 = \sigma_1.g^{P_1} = H(ID_B||b_1).g^{P_1} \bmod N$, $MB_2 = \sigma_2.g^{P_1} \bmod N = H(ID_B||b_2).g^{P_1} \bmod N$ and sends $MES_2 = \{MB_1, MB_2, \cdots, MB_k\}$ to Alice.

Encoding Phase:

M4. Alice chooses an arbitrary number R_a and computes $Z_A = g^{e.R_a} \bmod N$, $MB_k^* = (MB_k)^{e*R_a} = \{{MB_1}^{e.R_a}, {MB_2}^{e.R_a}, \cdots, {MBm}^{e.R_a}\} = \{(H(ID_B||b_1).g^{P_1})^{e.R_a}, (H(ID_B||b_2).g^{P_2})^{e.R_a}, \cdots, (H(ID_B||bm).g^{Pm})^{e.R_a}\}$.

M5. Alice performs arbitrary permutation $RPA = \zeta\{a_1, a_2, \cdots, ak\}^{R_a} = \zeta\{{a_1}^{R_a}, {a_2}^{R_a}, \cdots, {a_k}^{R_a}\}$ and sends $MES_3 = \{Z_A||MB_k^*||RPA\}$ to Bob.

M6. Bob also opts an arbitrary number R_b and computes $ZB = g^{e.R_b} \bmod N$, $(MES_1)^{e.R_b} = \{M1^{e.R_b}, {M_2}^{e.R_b}, {M_3}^{e.R_b}, ...{M_k}^{e.R_b}\} = \{(s_1.g^{R_1})^{e.R_b}, (s_2.g^{R_2})^{e.R_b}, \cdots, (s_k.g^{R_m})^{e.R_b}\}$.

M7. Bob chooses an arbitrary permutation $RPB = \zeta\{{b_1}^{R_b}, {b_2}^{R_b}, \cdots, {b_k}^{R_b}\}$ and sends $MES_4 = \{Z_k||(MES_1)^{e.R_b}||RPB\}$ to Alice.

6.2 Set Intersection Phase

M8. Alice computes $M_1 = ID_A \oplus h(MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$, $M_2 = h(ID_A \,||\, MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$ and forwards $\{M_1, M_2\}$ to Bob.

M9. Bob computes $M_3 = ID_B \oplus h(MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$, $M_4 = h(ID_B \,||\, MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$ and forwards $\{M_3, M_4\}$ to Alice.

M10. On receiving $\{M_3, M_4\}$ from Bob, Alice achieves ${ID_B}^* = M_3 \oplus h(MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$, computes $M_4^* = h({ID_B}^* \,||\, MES_1 \,||\, MES_2 \,||\, MES_3 \,||\, MES_4)$ and compares the computed M_4^* with the received M_4. If both are equal Alice authenticates Bob. Similarly, Bob achieves ID_A* from M1, and computes $M_2^* = h({ID_A}^*||MES_1||MES_2||MES_3||MES_4)$. If computed M_2^* equals the received M_2, Bob authenticates Alice.

M11. Alice share her random number to Bob by sending an encrypted message using the Bob public key, so that the message can be decrypted only by Bob using his private key, i.e. d_B. Its $D_{d_B}(E_{e_B}(ID_A||ID_B||R_a||R_1)) = \{ID_A, ID_B, R_a, R_1\}$.

M12. Similarly Bob also shares his arbitrary numbers by sending an encrypted message using Alice public key, i.e. e_A, i.e. $E_{e_A}(ID_B||ID_A||R_b||P_1) = \{ID_B, ID_A, R_b, P_1\}$.

M13. Alice computes a random permuted list $KA = \zeta A\{h(a_1||R_1)^{R_aR_b}, h(a_2||R_1)^{R_aR_b}, \cdots, h(a_m||R_1)^{R_aR_b}\}$ and direct to Bob. Bob also computes $KB = \zeta B\{h(b_1||P_1)^{R_bR_a}, h(b_2||P_1)^{R_bR_a}, \cdots, h(b_k||P_1)^{R_bR_a}\}$ and send it to Alice.

M14. In order to know the actual common attributes, Alice sends her random permutation by encrypting with the Bob public key, i.e. $e_B, E_{e_B}(\zeta A)$. Similarly, Bob sends his random permutations to Alice by encrypting with the Alice public key, i.e. $e_A, E_{e_A}(\zeta B)$.

Recovery Phase:

M15. Alice already knowing ζB, can able to compute ζB^{-1} and retrieves $\{h(b_1||P_1)^{R_bR_a}, h(b_2||P_1)^{R_bR_a}, \cdots, h(b_k||P_1)^{R_bR_a}\}$, similarly Bob able to compute ζA^{-1} and recover $\{h(a_1||R_1)^{R_aR_b}, h(a_2||R_1)^{R_aR_b}, \cdots, h(a_m||R_1)^{R_aR_b}\}$.

M16. For each attribute $\{a_1, a_2, \cdots, a_m\}$, Alice computes $\{h(a_1||P_1)^{R_aR_b}, h(a_2||P_1)^{R_aR_b}, \cdots, h(a_m||P_1)^{R_aR_b}\}$ and compares with the attribute list $\{h(b_1||P_1)^{R_bR_a}, h(b_2||P_1)^{R_bR_a}, \cdots, h(b_k||P_1)^{R_bR_a}\}$. The comparison gives the Alice, the number of attributes in common and their actual values with Bob. Bob also perform same computations as Alice. As Alice and Bob uses hash function and session specific arbitrary numbers to compute $\{h(a_1||R_1)^{R_aR_b}, \cdots\}$, $\{h(b_1||P_1)^{R_aR_b}, \cdots\}$, if an attribute sent by Bob is not matching against any value in the Alice attribute list, it is computationally infeasible for Alice to achieve or compute the non-matching attribute, due to one way property of hash function, even the Alice knows P_1, R_a, R_b. Similar is the case with Bob.

M17. Hence in our scheme, there is no chance of leakage of attributes to opponent, in case of non-matching attributes.

7 Informal Security Strengths of The Proposed Scheme

7.1 Resists Malicious and Semi-Honest Participant Attack (Attribute Verification)

In our proposed scheme, in Attribute Certification phase, the initiator Alice and the participant Bob submit their attribute set $a = \{a_1, a_2, \cdots, a_m\}$ and $b = \{b_1, b_2, \cdots, b_k\}$ to CA. The CA certifies the attributes and returns $A = \{(a_1, s_1), (a_2, s_2), \cdots, (a_m, s_m)\}$ to Alice, where $s_i = H(ID_A||a_i)^d \bmod N$. Similarly for Bob, CA returns $B = \{(b_1, \sigma_1), (b_2, \sigma_2), \cdots, (b_k, \sigma_k)\}$ where $\sigma i = H(ID_B||b_i) \bmod N$. As CA binding the attributes with their hash value, the participants are restricted to change their attributes later. This step restricts the attacks by malicious and semi-honest participants.

7.2 Resists Malicious Participant Attack (Attribute Mapping) Scenario 1

In matchmaking phase of our scheme, i.e. M5, M7 the initiator Alice sends the randomly permuted attribute set, i.e. $RPA = \zeta\{a_1, a_2, \cdots, ak\}^{R_a} = \zeta\{{a_1}^{R_a}, {a_2}^{R_a}, \cdots, {a_k}^{R_a}\}$ to Bob. Similarly Bob also opts an arbitrary number R_b and computes an arbitrary permutation $RPB = \zeta\{{b_1}^{R_b}, {b_2}^{R_b}, \cdots, {b_k}^{R_b}\}$. Due to the random

permutations, even though the participant or malicious attacker achieves a_1R_a etc, it is impossible to map ${a_1}^{R_a}$ to an entry in the list $\zeta\{{a_1}^{R_a}, {a_2}^{R_a}, \cdots, {a_k}^{R_a}\}$. Also in M11, M12 the Alice and Bob exchange their random numbers by encrypting with the public key of the opponents. In M11 Alice share her random number to Bob by sending an encrypted message using the Bob public key, so that the message is decrypted only by Bob using his private key, i.e. d_B. its $D_{d_B}(E_{e_B}(ID_A||ID_B||R_a||R_1)) = \{ID_A, ID_B, R_a, R_1\}$. Similar is the case with the Bob. Hence, it is impossible for an attacker to achieve the attributes of the participants.

7.3 Resists Malicious Participant Attack (Dynamic Attributes) Scenario 2

In all the previous works including Sarpong et al, the initiator and the participants make their attribute set random by exponentiating the attributes with random number. If the random numbers are known to the malicious users, they can retrieve the attribute values which are static. Hence, it will leak the attribute information. In our proposed scheme, Alice computes a random permuted list $KA = \zeta A\{h(a_1||R_1)^{R_aR_b}, h(a_2||R_1)^{R_aR_b}, \cdots, h(a_m||R_1)^{R_aR_b}\}$ in which a hash of an attribute is concatenated with a random number and exponentiated. In this case, the same attribute value results in a different hash value each time it is sent. Hence, it is difficult for an attacker to achieve any information from the attribute set.

Due to space restrictions, we have discussed above attacks only. Our scheme resists all major cryptographic attacks and achieves attribute privacy.

8 Formal Security Strengths of The Proposed Scheme

We prove the security strengths of our proposed scheme by comparing what a malicious attacker can do in the real protocol execution against what the attacker can do in an ideal world. In the ideal-world execution, both participants would submit their attribute set to an imaginary trusted certificate authority i.e. TCA. The trusted TCA certifies the attributes submitted. Once the validations are done, the communicating parties compute the intersection set. If a protocol participant submits a message without proper validation from TCA, the other participants ignore or drop the message. Automatically, this confirms that the real-world attribute set intersection protocol is as secure as the protocol in the ideal world that depend on TCA.

We now formally outline the ideal functionality. The security definition involves the communication between TCA and malicious attackers.

Authorize: If TCA receives an authorization or verification request from participant P_i, TCA computes $\sigma_i = H(ID_i||b_i)$ where $1 <= i <= k$ for totally 'k' attributes and submits σ_i back to P_i.

Set Intersect: Initiator P_i sends a request to perform set intersection to party P_j. Similarly P_j sends a request to perform set intersection to party P_i. P_i and P_j now run an ideal set intersection protocol as below:

1) P_i sends a set S_i to P_j and P_j sends S_j to P_i. On receiving the entities set, both P_i and P_j checks whether each attribute in S_i and S_j has proper validations from TCA. Let ${S_i}^* \leq S_i$ and ${S_j}^* \leq S_j$ denote maximal subsets of S_i and S_j that have proper validations.
2) P_i and P_j compute the intersection set $I \leftarrow {S_i}^* \bigcap {S_j}^*$.

8.1 Formal Security Analysis

In this part, we demonstrate the security strengths of our scheme formally by using the random oracle model and we will illustrate that our scheme is strongly secure.

In the random oracle model, an ideal simulator 'S' is constructed and given a black box access to an attacker 'E'. The communication between an attacker 'E' and the simulator 'S'go through only via oracle queries that models attacker 'E' competence in a real attack. To break the security strong point of the private set intersection protocol, 'E' simulates subsequent queries.

Simulation of different random oracles:

Lemma 1. *Assume that the DDH (Decisional Diffie Hellman hypothesis) assumption holds for exponentiation, and hash function 'H' behaves like a random oracle, then the proposed hybrid protocol securely performs the 'Set Intersection' function described above.*

8.1.1 Simulation of Hash Query

Simulator 'S' maintains an initial empty hash list ${L^{List}}_h$ for the hash function h. The List maintains a tuple (x,P). On receipt of the hash query for an input 'x', 'S' will do a lookup operation.If the result exists, returns the same answer, else, it generates a random number g' ϵ G and returns g'. S' inserts (x,P) into the List.

8.1.2 Simulation of Authorize Query

Simulator 'S' on receiving the authorization or validation request i.e. to sign an element 'x' on behalf of certificate authority 'i' for a corrupted participant P_A (controlled by an attacker 'E'), 'S' makes a hash query on input (x, P_A) and on determining the hash value, 'S' computes the signature and returns the same to an attacker 'E'. (The simulator 'S' knows all the signing keys of all the protocol participants).

Table 1: Comparison of security features

Attacks/Protocols	Ours	[4]	[13]	[12]	[15]
Resists Semi-Honest Attack	Y	Y	Y	Y	Y
Resists Malicious Attack (Scenario 1)	Y	Y	Y	Y	N
Resists Malicious Attack (Scenario 2)	Y	N	N	N	N

Table 2: Comparison of complexity

Protocols	Computational Complexity	Communication Complexity
[10]	$O(m \log\log n)$	O(m+n)
[8]	O(R^2.n)	O(n^2)
[11]	O(R^2.n)	O(n.R)
[13]	2(N-1)(m+n)PM+2(N-1)DH	N-1)(m+n+5)
[12]	2(N-1)(m+n)PM	(N-1)(m+n+4)+6
[15]	2+m(n+1)+k(N+1)PM+(m(2+N)+n(2N+m+2)+2)EXP+(3N+1)Enc	O(m*n)
Proposed	2+m(n+1)+k(N+1)PM+(m(2+N)+n(2.N+m+2)+2)EXP+(3N+1)Enc	O(m*n)

8.1.3 Simulation of Set Intersection Query

Whenever an attacker 'E'submits a request to perform the set intersection protocol, S performs the following simulation. Assume that 'E'is imitating Alice as discussed in the above section.

'E' chooses an arbitrary random number some $A_1 \epsilon G$ and sends a set of encodings $MAS_1 = \{MA_1, MA_2, \cdots, MA_m\} = \{H(ID_A||a_1)^d.g^{A_1}, \cdots\}$ to S. S also chooses an arbitrary number B_1 computes the encodings, i.e. and directs $MBS_1 = \{MB_1, MB_2, \cdots, MB_k\}$ to 'E'. 'E'chooses an arbitrary number RA and computes the encodings $Z_A = g^{e.R_A} \bmod N$, $MBS_1{}^* = (MBS_1)^{e*R_A}$ = and submits $\{MB_1{}^{e.R_a}, MB_2{}^{e.R_a}, ...MB_m{}^{e.R_a}\}$ to S. 'E' performs arbitrary permutation $RPA = \zeta\{a_1, a_2, \cdots, a_m\}^{R_A} = \zeta\{a_1{}^{R_A}, \cdots, a_m{}^{R_A}\}$ and directs $MAS_2 = \{Z_A||MBS_1{}^*||RPA\}$ to S. 'S' chooses an arbitrary permutation $RPB = \zeta\{b_1{}^{R_B}, b_2{}^{R_B}, \cdots, b_k{}^{R_B}\}$ and sends $MBS_2 = \{ZB||MAS2^*||RPB\}$ to 'E'. It is not hard to see that S can compute all encodings, as it knows the secret signing keys ski for all the participants.'E'and S shares the random numbers used, i.e. A_1, B_1. Finally S and 'E' computes the intersection set.

It is clear that, except the attacker 'E' is able to fake or forge an encoding for an attribute, it does not possess a proper signature, then the joint output of all participants in the ideal world are identically distributed as similar to the proposed protocol. Assume that if 'E' did fake or forge an encoding for some element 'bi' which it is not validated or authorized by the TCA, then in the ideal world the protocol participants will filter out that attribute from the resulting set intersection, which results in the output distribution to be different in the ideal protocol from the proposed one.

9 Simulation and Experimental Evaluation

In this segment, we scrutinize the computational complexity of proposed and various related schemes through simulations.

9.1 Complexity Analysis

The computation cost is calculated based on the number of resource consuming 1024-bit multiplication, 1024-bit exponentiation and SHA-160 hash operations on mobile devices. The communication overhead is computed by the number of bits transmitted and received.

Table 1 confirms that our proposed scheme resists all major attacks both passive and active.

In Table 2, PM denotes a power modular; R denotes number of rounds; m, n denote number of Alice and participants attributes; EXP denotes an exponential operation; Enc denotes an encryption; N denotes number of participants.

Table 2 confirms that our scheme requires similar computational complexity compared to [7, 12] but negligibly higher complexity compared to the traditional schemes [8, 10, 11, 13]. But the overhead is perfectly valid, due to its security strengths. As discussed in the system architecture, we simulated our proposed scheme with a Samsung Galaxy J7mobile device consist of 1.5 GHz CPU. The simulation code is written in Java. We have considered the users $N = 5, 10, 20, 30, 40$ and each user is considered to contain varying attributes $k = 5, 10, 151$ (See Figure 2).

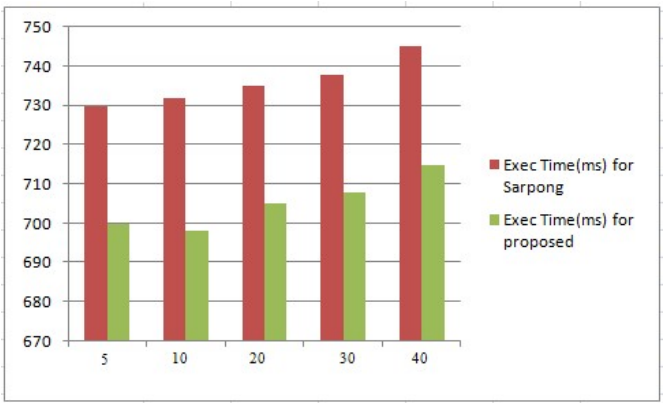

Figure 2: The simulation

10 Conclusion

The involvement of user's specific and sensitive data in MSN demands for a light weight and secure matchmaking algorithm, which resists attribute leakage to participants. Sarpong et al. had proposed first of its kind of matchmaking algorithm which selects the participants that contains the threshold level of attributes matching. We have cryptanalyzed Sarpong et al. scheme, and demonstrated that their scheme fails to achieve attribute privacy and requires huge storage and computation cost. We have proposed an efficient algorithm, which resists the pitfalls found in Sarpong et al. algorithm and other related schemes (static attribute representation). We also conducted experimental analysis of our scheme and illustrated the results.

References

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

[2] S. Y. Chiou and C. S. Luo, "An authenticated privacy-preserving mobile matchmaking protocol based on social connections with friendship ownership," *Mathematical Problems in Engineering*, vol. 2014, Article ID 637985, 2014.

[3] X. Dong, "A multi-secret sharing scheme based on the CRT and RSA," *International Journal of Electronics and Information Engineering*, vol. 2, no. 1, pp. 47–51, 2015.

[4] A. Evfimievski, R. Agrawal, and R. Srikant, "Information sharing across private databases," in *Proceedings of ACM SIGMOD*, pp. 86–97, 2003.

[5] L. Guoy, J. Liu, R. Hao, B. Yangx, S. Jiang, X. Zhu, "Efficient private matching based on blind signature for proximity-based mobile social networks," in *IEEE International Conference on Communications (ICC'15)*, pp. 3246–3251, 2015.

[6] S. Huang, S. Griswold, K. Li, T. Sohn, "People-tones: A system for the detection and noti cation of buddy proximity on mobile phones," in *Proceedings of 6th IntConfon Mobile Systems (MobiSys'08)*, pp. 160–173, 2008.

[7] Y. H. Huang, S. H. Chiou, "Mobile common friends discovery with friendship ownership and replay-attack resistance," *Wireless Networks*, vol. 19, pp. 1839–1850, 2013.

[8] L. Kissner and D. Song, "Privacy-preserving set operations," in *Advances in Cryptology (CRYPTO'05)*, LNCS 3621, pp. 241–257, Springer, 2005.

[9] M. Li, N. Cao, S. Yu, W. Lou, "Findu: Privacy-preserving personal profile matching in mobile social networks," in *Proceedings of IEEE INFOCOM*, pp. 2435–2443, 2011.

[10] K. Nissim, M. Freedman, and B. Pinkas, "Efficient private matching and set intersection," in *International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 1–19, 2014.

[11] 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.

[12] Y. Wang, T. Zhang, H. Li, L. He, and J. Peng, "Efficient privacy preserving matchmaking for mobile social networking against malicious users," in *IEEE 11th International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 609–615, 2012.

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

[14] D. Xing, Y. Fang, H. Lin, S. S. M Chow and Z. Cao, *Privacy Preserving Friend Search over Online Social Networks*, Cryptology ePrint Archive, 2011. (`http://eprint.iacr.org/2011/445.pdf`)

[15] X. Zhang, S. Sarpong, C. Xu, "An authenticated privacy-preserving mobile matchmaking protocol based on social connections with friendship ownership," *International Jpurnal of Network Security*, vol. 17, no. 3, pp. 357–364, 2015.

Biography

K. Arthi have been working as a Associate professor of Department of Information Technology in VelTech Technical University, Avadi. He have completed my Ph.D in the area of Wireless Sensor Networks in Anna University, Chennai during 2015 and my ME (Embedded system technology) also from the same university during 2005. He completed my BE (IT) in Periyar University during 2002. He have published many National and International Journal Papers.

M. Chandramouli Reddy is a Research(Ph.D) Scholar, Department of Computer Science and Engineering in VelTech Technical University, Avadi. He have completed his

M. Tech (CSE) in JNT university, Hyderabad during 2006 and he completed his BE (CSE) in Madras University, Chennai during 1999. He have published many National and International Journal Papers.