International Journal of Engineering Science Invention
ISSN (Online): 2319 – 6734, ISSN (Print): 2319 – 6726
www.ijesi.org ||Volume 4 Issue 5 || May 2015 || PP.07-17

Distributed Privacy Preservation Matchmaking protocol in Mobile Social Networks

John Bosco Aristotle Kanpogninge Ansuura[1], Qi Xia [1,]Benjamin Klugah-Brown[2]
School of Computer Science and Engineering, University of Electronic Science and Technology of China, 611731, Chengdu, China

ABSTRACT : *Friend discovery based on common attributes or similarity of users' profile is a key component of mobile social networking. Preserving the privacy of profiles of matching users whiles matchmaking is the major challenge of this system. These profiles contain very sensitive information such as interests, political tendency and health conditions. Matchmaking protocols have been based on the assumption that users of the system must agree to some compromise with group managers, setup procedures and revocation procedures. It is however disturbing that these users desire systems with no trusted third parties and a complete unconditionally user ambiguity.*

To this end, we are have designed a distributed system based on Xie et al that will preserve user privacy, maintain the Security of users, eliminate the trusted third party and enhance the efficiency. Our focus is to design a distributed privacy preservation matchmaking protocol aimed at privacy preservation and security of users with a reduced communication cost.

KEYWORDS-*Privacy-preserving, RSA signatures, Matchmaking, mobile social networks.*

I. INTRODUCTION

Mobile Social Networks (MSN) in recent times provide us with personalized services for convenience in terms of portability and mobility of mobile devices and services. This area has gained much attention from researchers of many different fields. This is because all entities (e.g. people, devices, or systems) in the world are related to one another in one way or the other. Whiles Mobile networks provide the mobility for users and mobile devices, social networks provide the social connection to these users via the mobile devices which run mobile social network applications. The most common examples of Social Networks are Myspace, Weibo, Facebook, LiveJournal, Twitter and Flickr. Social networks are popular ways of interaction among people and devices. The Deployment of MSN make use of what are referred to as user interests and profiles. These interest are similar to what most online social networks use. For example, similar to most of the online social network applications, a mobile social network user is expected to freely search its potential common-interest friends by matching his/her *interests* with the *personal profiles* of the searching targets rather than making the profile matching directly.

Assuming Alice has her personal profile, which include three attributes: age, girl and movie. She is interested in finding a boy with similar age and hobbies. Conversely, Bob also has his own profile and interests. A successful matching could be achieved such that Alice's profile matches Bob's interest whiles at the same time Bob's profile matches Alice's. These attributes are used to establish friend discovery and fairness matchmaking operation. Social networking sites enable users not only to communicate with existing friends but also people can find and make friends with other people who have common interest such as from same school, same company, age etc. In MSN, users not only find and make new friends using the features of the traditional social networking sites, but also can find and make friends using the geographical distance between two users which is an extra matchmaking criteria. With mobile phones users, the chance of meeting friends and strangers as one walk around is very high. When two mobile phones are geographically nearby, a matchmaking operation takes place and detects common interests of the devices' owners. If a match is found, the devices notify their owners, who can immediately meet each other in person. Matchmaking is key attribute of mobile social networking where users of the product find friends only by sharing common attributes. The worldwide mobile phone usage have grown from 12.4 million to over 6.9 billion between the periods of 1990 to 2014. Facebook, a popular online social networking site has over 800 million active users [1].

Privacy and ensuring of security of users whiles matchmaking are the key challenges of Mobile Social Networking. A lot of architectures and matchmaking protocols have been proposed for MSN. We examine several existing architectures and matchmaking protocols in MSN. A distributed architecture with security and privacy preservation can be the most suitable choice for MSN as users are generally not ready to compromise but yet desire to find new friends whiles maintaining their privacy and security.

Three main architectures have been proposed and exist in the MSN paradigm.

These are centralized, distributed and Hybrid architectures. In the Centralized MSN approach, users simply rely on a server and with the knowledge of users' current location and attributes the server performs the matchmaking on their behalf. Social serendipity [2], and PeopleTone[3] are some popular examples. In Social Serendipity [2] a server is involved in every matchmaking operation. Users will be notified when they are nearby and the similarities of their profiles exceed a threshold. This is a limitation to those systems as this makes it vulnerable, for the server can learn the interest of the users thereby making it difficult for the server to detect which two users are close by and having a matching interest. Though easy to implement, not all users are willing to submit their personal information to the server because of the privacy concerns. The trusted server on the other hand is likely to be the bottleneck of the application in MSN, and the one-point failure problem has to be considered. Additionally, servers learn the attributes of users and also servers are generally based on the connection of the internet which is equally costly. In Fully distributed MSN approach, users broadcast their personal information to the network, for example using Bluetooth/Wi-Fi to any Bluetooth/Wi-Fi device nearby. Each user performs the matchmaking to find their intersection set when receiving attributes from other users. This is more improved than the centralized architecture as it obviates the need of having to access a server [29]. However, users broadcast more personal information than necessary, this revealed that it cannot prevent malicious users from acquiring extra information of other users. FindU[4], E-smallTalker[5], mobiclique[6] and Agrawal *et al* [7] uses this approach. Agrawal *et al* [7] proposed a distributed preserving matchmaking protocol by applying the commutative encryption that gives the same result with two different private keys despite the order of encryption. Thus users that participate in the matchmaking exposes to each other only the common attributes. But users' profiles are not certified in this protocol so that a malicious user has a chance to freely choose the inputs to the protocol to get more information than intended. The work in [8] proposed a private matching using certified attributes to defend against malicious users. This protocol mandates users to attach a user ID to attributes and then signed by the trusted third party before the matchmaking to prevent impersonation and freely choosing attributes by malicious user. In order to maximize privacy of the attributes by only exposing common interests to one or few numbers of users, best matchmaking protocol based on multi-party computation is proposed in [4]. Best match selection based on two party computation is also proposed in [9]. In both cases the best match is selected based on the number of common attributes the users share. Lastly, in Hybrid MSN approach, a trusted server is only needed for the purpose of management and verification, and it not involved in the matchmaking phase. This architecture is easy to manage, and effective in guaranteeing the security of the matchmaking.

The improvement of efficiency whilst maintaining user privacy and security is now equally a challenging issue in MSN. Matchmaking protocols can maintain user privacy and security but with high communication and computation overhead. The issue of mobile device storage and battery life consumption is major factor in MSN.

MSN applications run social network applications on mobile devices and also making it possible for users to be mobile yet socially connected.

The main significance of our research work is that we have provided a distributed privacy preserving matchmaking protocol based on the work of [8] with a reduced communication and computational overhead.

II. RELATED WORK

Matchmaking protocols can be divided into three categories according to personal information collection, matched attributes computation and results dissemination. Based on this, we give a survey presenting each of the criteria stated above. In the earliest approach, services of a trusted central server were employed, which is directly involved in each step of the matchmaking process. They were basically internet-based. That is to say, the central server collects users' attributes and location information, computes the match and notifies the best match to the initiator. Most matchmaking protocols such as Social Serendipity [2], a server is involved in every step of the matchmaking operation. In this system, Bluetooth MAC addresses are mapped to users' profiles on other social networking websites. This facilitated face-to-face interaction between nearby strangers, it retrieves their mobile devices' Bluetooth Mac addresses and uses them to retrieve the strangers' profiles on the server for similarity matching. This system uses SMS for device-server communication. Users will be notified when they are nearby and the similarities of their profiles exceed a threshold. This is a limitation to those systems as it makes them vulnerable for the server to learn the interests of the users and thereby making it difficult for the server to detect which two users are closed by and having a common matching interest.

Due to this limitation, a variant of the centralized server MSN was proposed aimed at enhancing awareness and interacting geographical location and notifying its nearby friends. This gives friends knowledge of each other's whereabouts, which facilitates opportunistic interactions. PeopleTone[3], is one of such application used for "nice to know" contextual information like the proximity of friends, and making users aware of such information. This research explored proximity detecting, sensor noise and power consumption reduction and peripheral cues generation.

The system measures the precision and recall of our proximity detection, identification of the correspondence of vibrations to music clips which conveys buddy proximity via peripheral cues that are uniquely assigned to buddies. The central server system was not without drawbacks. Some of these concerns were; (1) users are naturally unlikely willing to send their personal information to the server. (2) The centralized server is generally based on the connection of the internet, in some application scenarios, users would like to perform matchmaking through multiple communication channels (e.g. Bluetooth/Wi-Fi) (3) one-point-failure and bottle-neck problems limit the systems' scalability. (4) Communication between server and devices (via SMS, Wi-Fi, or 2G/3G) may be costly, unreliable and even unavailable. (5) Lastly user's privacy maybe compromised, e.g. by saving location and other personal data on a third-party server. Mechanisms have to be involved to provide security protection of the centralized server. On the other hand, with the number of users increasing, the centralized server may become overloaded which may lead to quality of service (QoS) dropping.

Later, the distributed mobile social networks was proposed for matchmaking. In this architecture, mobile devices are allowed to directly communicate with each other without requiring a trusted server. These protocols partially eliminated the total involvement of servers in the operation by introducing a middle ware that allows mobile phone users to connect over ad-hoc networks. An example of such are the MobiClique[6], FindU[4] and SmallTalker[5]. Mobiclique[6] is a middleware that allows mobile phone users to connect to others over ad-hoc networks to exchange social network identity information and forward messages. It was an improved version of Social Serendipity [2] where the server is removed from the matchmaking process. In this system there is a server, Facebook, which assigns identifiers to the users. It allow users to store their profile information on their smart devices and perform profile exchange with the in neighbors using Bluetooth. A friendship is created based on the user's profiles acquired. This application does not put into consideration an adversary attack. It assumes that all users are trusted, and ignores privacy and security. This means that anyone in range can intercept the information and perform a mischievous attack with it. The smallTalker proposed by Yang *et al* [5], is a practical system for matching people's interests before initiating a small-talk. This however reveals the exact common attributes between initiator and every other user, which could be more than necessary. "FindU" [4] provides privacy-preserving personal profile matching services in a fully distributed architecture. The operations, such as the distribution of personal attributes data, the computation of the intersection set, and the dissemination of results are performed among multi-parties without any trusted third party. The attributes of the initiator and the candidates are shared among multi-parties using Shamir Secret Share Scheme (SS), the computing of common attributes set are conducted among multi-parties as well. The entire procedure requires neither Internet access nor a centralized server, which reduce the system cost. However, such fully distributed systems are not convenient to be managed. The efficiency of the system may get worse when the users number becomes too large. We are adopting a fully distributed MSN mechanism to privacy-preserving attributes.

Lastly, hybrid MSN mechanism was invented, where a trusted centralized server is needed only for the purpose of management and verification, and it does not participate in the matchmaking operations. This mechanism can provide efficient matchmaking services with the relatively high scalability. Xie*et al*[8], Li *et al*[10] and Yong Wang *et al* [9] used this mechanism in their work for privacy-preserving matchmaking protocols.

In Xie*et al* [8], which is the reference point of this paper, the identify signer assigns an identity and certificate to identify each user. It assigns a global ID using either the Bluetooth address or the IMEI of the mobile device for verification. The user then determine a number of interest to find a friend to prevent users from detecting our information. The users send their interests to a personal interest signer who signs them to help with authentication of certifications, certifying that it doesn't emanate from an arbitrary interest. Users create lookup names with an ID assigned by the PIS for each interest. The initiator of protocol will have to pair Bluetooth and run signature-based authentication protocol to authenticate each other. The matchmaking protocol is made up of fourteen lines of communication. If a user is found to have cheated his signature would be revoked and the user ID blacklisted. These kind of protocols use the fairness aware friend discovery protocol which involve using the private set intersection protocol to find the common interest and these have been working fine. As most of these system must be ran twice before users know the common interest, the initiator may abort the protocol after the friend is discovered. This is described as run-away attack.

III. PROBLEM STATEMENT

A. System model

This model describes what goes on in matchmaking in mobile social networking. The components making our system model are Users, Mobile devices and Identity Signer as shows in Figure 1. This consist of a two phases, the initial and matchmaking phases in that order.

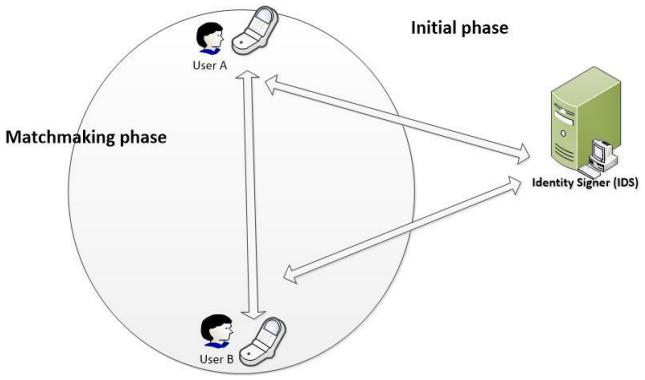

Figure 1:System Model

Users U: U= $\{U_a, U_b, U_c \ldots\}$ is a set of registered users, and each user $U_i \in U$ is equipped with a mobile device. Some users maybe malicious trying to infer other information on the system by utilizing the vulnerabilities. There are also semi-honest users trying to learn others' private attributes from the messages sent to them. They determine attributes and compute the RSA signature.

Mobile Device: Each device has its owner's attributes set configured prior to the matchmaking. The security of these devices are sole responsibility of the owners. $X_i = \{X_a, X_b, X_c \ldots\}$be the attributes of X_a^ieach $X \in X_i$, is an attribute of U_i. When a device is near to another device, they communicate directly with each other (e.g., by Bluetooth/Wi-Fi) and compute the size of the intersection set.

Identity Signer (IDS): An ID signer is a trusted third party (TTP) which assigns an identifier and a certificate to identify each user. This TTP learns about identity (ID) information only. To identify and authenticate a user, there should be a TTP. A globally unique ID is assigned to each user and this is done once. The Bluetooth address could be used as an identifier for each Bluetooth device because it is free and globally unique. However, it is proven that the Bluetooth address can be a modified hardware, change or Modify Bluetooth Device hardware (MAC) address [45]. IMEI is an identifier for a mobile phone. Theoretically, IMEI can be used as a User's ID, but so far, we have not found any literature using IMEI in such a way.

One possible reason is that it is not easy for a mobile device to validate a received IMEI. This could be addressed if the telecom companies issued IMEI certificates to their customers. If that is the case, the telecom companies are our ID signers. In short, we need an ID signer to verify users' personal information and issue one User ID and one certificate of the User ID for one user. For example, a trusted certificate authority or CA (e.g., VeriSign) can act as an ID signer. A CA issues a digital ID for a user, including a public key, name and email address, name of the CA, serial number of the digital ID, digital signature of the CA, and so on. Usually, it costs money to register a digital ID, but people could use the digital ID for other applications, such as secure email. In general, the CAs do not guarantee that they assign only one user ID for each person/device, but this can be done if they co-operate with us. The CAs usually have detailed information about their customers (sufficient to identify a person). For example, if a customer requests a user ID and the certificate for our system (a CA can just include any special label in a normal certificate to distinguish the certificates for our system from the normal ones), the CAs can check the history of this customer and refuse to issue a new user ID for them if they already has one [46].

Lastly the IDS signs the interests using it master key. This will help authenticate that the user's interest has been certified and it is not emanating from an arbitrary interest by using

$S_j = \{sig_d(S_i \parallel ID_j)\} = \{S_{a_1}, \ldots, S_{a_{max}}\}, where\ j\ is\ user\ index$and $i \in Attr_j$.

B. Adversary Model

The adversary considered in this work is same asXie*et al* [8]. The adversary thwarting the system can be classified into two categories according to the behaviors they conduct to obtain extra information: Semi-honest adversaries (or curious adversaries) are entities in the system that follow the protocol properly, with the only exception that an adversary may keep a record of all intermediate computation and communication in order to find extra information than intended for him. Malicious adversaries can deviate from the designated protocol, change their input, halt the protocol run before finishing and they will try to obtain the most extra information to other parties by providing false inputs. We assume user trust the match selected to share his formation. Even if it is difficult to say the protocol is immune to malicious attacks completely we tried to design protocols that defend malicious attacks we mentioned and other related papers. In particular, we consider the following adversary model:

1. Getting users interests without getting caught for cheating unless they actually have the same interests.
2. Exploring users' interests by including all likely or a large number of prevalent elements in their interest set (brute force attack). We allow users to create only a limited number of interests, maximum of ten.
3. Impersonating other users. We ask each user to create a pair of asymmetric keys and use the hash value

of the public key as their user ID. After two users meet, they first exchange their public keys, and they then negotiate a secret key using each other's public key. They can derive a session key using this secret key. This authenticates each partner of the protocol.

4. Eavesdropping the communication between any two users. Sensitive information is encrypted by the session key that the two users established.

C. Security Assumptions

These are some threats we do not consider in our system and some assumptions that we made:

1. Users keep their private keys safe, so that malicious users could not steal their private keys to impersonate them.
2. The third party server is not compromised by attackers.
3. In our protocol, we assume that most users are rational and they are honest but curious. This means that most users are not going to reveal information if it brings them negative effects.
4. Users trust that their matched friends will not disclose their matched information.
5. Users will finish running the protocols once started.

D. Designing Objective

The proposed enhanced privacy-preserving matchmaking protocol for mobile social networks should satisfy the following objectives.

(1) Privacy-preservation: the proposed matchmaking protocol can preserve the user's privacy.

(2) Security: the proposed protocol's security should not be compromised and it shall maintain same Security as Xie's.

(3) Anonymity is an important form of privacy protection. In MSN users most especially prefer anonymous by making

(4) It difficult to distinguish participants from non-participants so as to maintain their privacy in participating the process.

(4) Trusted third party: fully distributed with no trusted third party involve in the matchmaking stage.

(5) Efficiency: the proposed protocol should be efficient. The setup, computational and communication overhead should be reduced to the barely possible cost.

IV. OUR PROTOCOL

This protocol is made up of seven algorithms, setup, Key and Attributes generation, attribute selection, verification I, Response, blind matching and verification II.

Matching phase: our matchmaking protocol

Setup $(K) \Longrightarrow p, q, g, G, N$

In this algorithm the IDS will input K, where K is the security parameters and output $N = p.q$ such that p and q are large prime numbers.

$S = \{s_1, s_2, \dots, s_n\}$is the universal set of attributes where n is number of attributes

θ is the minimum shared attributes for authentication between any two parties such that

$S_A \cap S_B \geq \theta$whereS_A, S_B are attributes of party A and party B respectively.

Key and Attributes Generation$(K, Attr) \Longrightarrow K = \{P_k, P_r, a, b\}, Attr = \{S_A, S_B, S\}$this algorithm is by the IDS, it will create private key P_r for each party which is transmitted through a secured channel, whereas the P_k is published for all.

(1) **Key Generation,** we follow RSA's step, IDS choose $e, public\ key$, at random and computes the $d, secret\ key$, such that $\gcd(e_i, \emptyset(N)) = 1$, where i is user index

(2) **Attributes Generation,** The IDS assign attributes for Alice and Bob according to their privileges. Attributed granted for Alice

1. Alice $S_A = \{S_i\}\ i \in [n], |S_A| = m \leq N$ and for
2. Bob $S_B = \{S_j\}\ j \in [n], |S_B| = n \leq N$

Attribute selection ($SK_i, Attr_i$)

This algorithm is run by user who wants to find some friends with same features or attributes at least equal to θ. some attributes $'Attr_i \subseteq Attr_i$ he wants from S_A from (1) and same with Bob form S_B each party chooses some random number.

3. Alice chooses $\alpha \in_R \mathbb{Z}_p^*$
4. Bob choose $\beta \in_R \mathbb{Z}_p^*$
5. Alice chooses $Attr_A = \| S_i^\alpha$, and send $sig_A = \{S_{a_i}^\alpha \left|\left| Sig_{Sk_A}(ID_A || S_{a_i}^\alpha)\right\}\right| g^\alpha\ i \in [Attr_A]$ to Bob

6. Bob choose$Attr_B = \| S_j^b$, and send $sig_B = \{S_{b_j}^{\beta} \left|\left| Sig_{Sk_B}\left(ID_A \right|\left| S_{b_j}^{\beta}\right)\right.\right\}\left|\right| g^{\beta}\ j \in [Attr_B]$ to Alice

Verification I (Pk_j, sig_j)

7. Alice verifies Bob **Verify(Pk_B, sig_B) if Ok** then go to next step else abort, same for Bob
8. Bob verifies Alice **Verify(Pk_A, sig_A)a if Ok** then go to next step else abort

Responses (sig_B, α, β)

In these steps after parties verification for integrity sig_A **and** sig_B from both parties, they will reply from each other.

9. Alice $resp_A = \{(S_{b_j}^{\beta}, (S_{b_j}^{\beta})^{\alpha})\}\ \ j\ \in [{}'Attr_B]$ and
10. Bob $resp_B = \{(S_{a_i}^{\alpha}, (S_{a_i}^{\alpha})^{\beta})\}\ \ i\ \in [{}'Attr_A]$
11. Alice computes $\delta_A = resp_A||sig_{Sk_A}\left(ID_B||resp_A||(g^{\alpha})^{\beta})\right\}||(g^{\alpha})^{\beta}$ and sent to Bob
12. Also, Bob computes $\delta_B = resp_B||sig_{Sk_B}\left(ID_A||resp_B||(g^{\beta})^{\alpha})\right\}||(g^{\beta})^{\alpha}$ and sends to Alice

Blind matching ($resp_A, resp_B, m$)

This algorithm accepts the $resp_i$, where i is the user index, as inputs such that
$common\ interest, m = |{}'Attr_A \cap {}'Attr_B|$ Using Private set Intersection protocol

13. If $\left|\{(S_{b_j}^{\beta})^{\alpha} \cap (S_{a_i}^{\alpha})^{\beta})\}\right| \geq \theta\{^{sucess\ ,k=(g^{\beta})^{\alpha}=(g^{\beta})^{\alpha}}_{terminate\ \ protocol}$

$$i \in {}'Attr_A$$
$$j \in {}'Attr_B$$

Verification II

In this algorithm the users establish a secured channel and each sends confirmation messages to verify this channel

14. Bob sends $m||\ sig_{Sk_B}(ID_A||resp_B||g^{a}|E_k(m||sig_{Sk_B}(ID_B||m))$to Alice
15. Alice $E_k(m||sig_{Sk_A}(ID_A||m))$Bob and the protocol is ended with the users who are in a secured channel.

A simplified version of the protocol is show below.

Let $Attr_A = ||S_i^{\alpha}$

1. Alice $\{S_{a_i}^{\alpha}\left|\left|Sig_{Sk_A}(ID_A||S_{a_i}^{\alpha})\right\}\right|\ g^{\alpha}\ i \in [Attr_A]$ Bob

Chooses $\alpha \in_R \mathbb{Z}_p^*$ chooses $\beta \in_R \mathbb{Z}_p^*$

Let $Attr_B = ||S_j^b$

2. Alice $\{S_{b_j}^{\beta}\left|\left|Sig_{Sk_B}\left(ID_A\right|\left|S_{b_j}^{\beta}\right)\right\}\right|\ g^{\beta}\ j \in [Attr_B]$ Bob
3. Alice verify Bob Verify (Pk_B, sig_B) and Bob verify Alice Verify (Pk_A, sig_A) . if ok then go to next step else abort.
4. Alice computes $resp_A = \{(S_{b_j}^{\beta}, (S_{b_j}^{\beta})^{\alpha})\}\ \ j\ \in [{}'Attr_B]$and Bob computes $resp_B = \{(S_{a_i}^{\alpha}, (S_{a_i}^{\alpha})^{\beta})\}\ \ i\ \in [{}'Attr_A]$.
5. Alice $resp_A||sig_{Sk_A}(ID_B||resp_A||(g^{\alpha})^{\beta})\}||(g^{\alpha})^{\beta}$Bob
6. Alice $resp_B||sig_{Sk_B}(ID_A||resp_B||(g^{\beta})^{\alpha})\}||(g^{\beta})^{\alpha}$Bob

$$m = |{}'Attr_A \cap {}'Attr_B|$$

$$\text{If } \left|\{(S_{b_j}^{\beta})^{\alpha} \cap (S_{a_i}^{\alpha})^{\beta})\}\right| \geq \theta\left\{^{sucess, k=(g^{\beta})^{\alpha}=(g^{\beta})^{\alpha}}_{terminate\ protocol}\right.$$

$$i \in m$$

$$j \in n$$

7. Alice$m||\ sig_{Sk_B}(ID_A||resp_B|g^{a}|E_k(m||sig_{Sk_B}(ID_B||m))$ Bob

Alice $E_k(m||sig_{Sk_A}(ID_A||m))$ Bob

A. Distributed Privacy preservation matchmaking protocol (DPPMP)

In this protocol we shall represent Alice's interest as $\{S_i\}$ (m is size of Alice's set) and Bobs interest is also represented with $\{S_j\}(n$ is the size of Bobs set) Let represent $(S_i)^a modp$ with$(S_i)^a$, length of string s with$l(s)$. L*et al* so assume that Alice and Bob agree on a value g. Alice and Bob will each determine and generate their own RSA signature.

Before running this protocol, Alice and Bob should pair their Bluetooth devices.

The first result is reported by the party who initiates the matching protocol on a common value which we will denote as(g). Before our protocol is activated several processes take place to lay the foundation for the matchmaking protocol with the help of IDS. These processes include each user's pre-determined number of attributes to use.

We are going to outline what goes on in each of our eight (8) steps protocol.

Firstly, in step 1 Alice chooses $\alpha \in_R \mathbb{Z}_p^*$, then sends her chosen attributes signed with credentials signed by the ID Signer along with g^α to Bob.

In Step 2, which is similar to step 3, Bob chooses $\beta \in_R \mathbb{Z}_p^*$, and then sends his chosen attributes signed with credentials signed by the ID signer along with a computed g^β.

In step 3, both Alice and Bob verify the signatures. If any failed to verify the others signature the protocol terminates. The verification of signatures is to identify and authenticate each party involved in the matchmaking process. This authentication is run to enable the parties involved to exchange their certificates belong to the group to prevent communicating with adversaries.

Subsequently in step 4, Alice computes the exponential values of the interests she received from Bob, further hiding the interests that is $resp_A = \{(S_{b_j}^\beta, (S_{b_j}^\beta)^\alpha)\}\ \ j \in ['Attr_B]$Alice then signs$resp_A$, attaches Bobs ID_B and sends it to Bob along with g^α and g^β.

At this stage Bob computes the exponential values of the interest using his chosen random number to get $resp_B = \{(S_{a_i}^\alpha, (S_{a_i}^\alpha)^\beta)\}\ \ i \in ['Attr_A]$ and verifies ID_B sent to Alice including g^β andg^α.

In step 5, Alice sends the computed $resp_A$ along with Bob ID signed with another signature, $resp_A \left|\left|sig_{Sk_A}\left(ID_B\middle|\middle|resp_A||(g^\alpha)^\beta\right)\right\}\right|\left|\ (g^\alpha)^\beta\right.$ to Bob. Bob must verify the new signature. The protocol terminates if signature fails to verify else he does the intersection and computes the common interest $m = |S_A \cap S_B|$. The protocol equally terminates $if\ m = 0$.

Similarly in step 6, Bob sends the computed $resp_B$ along with Alice ID sign with $resp_B||sig_{Sk_B}\left(ID_A||resp_B||(g^\beta)^\alpha\right)\}||(g^\beta)^\alpha$ to Alice. Alice must verify the new signature. The protocol terminates if signature fails to verify else Alice also computes $m = |S_A \cap S_B|$ and she makes sure $m = 0$else the protocol terminates. Bob can then compute the session key K i.e. $k = (g^\beta)^\alpha$

In step 7, Bob sends the following together to Alice that is his computed interest $resp_B$, signed by him, encrypted value of his common interest computed m along with his ID_B and common interest authenticated by the Alice also compute for k at this stage this is done to establish a successful connection and prevent cheating.

In step 8 Alice sends his computed interest m, and the common interest and Alice *ID* encrypted using the session key, and send to Bob.

In steps 7 & 8 the messages are not sent in plain text but encrypted using authenticated Diffie-Hellman [43] with the aid of the secret key negotiated between the two parties.

V. SECURITY ANALYSIS AND PERFORMANCE EVALUATION

This section of the paper deals with the performance and security analysis of the proposed system model and protocol. It takes in consideration the performance of the protocol with reference to overhead cost and how secured the proposed system is against malicious users.

A. Security Analysis

The security of this protocol holds if and only if Agrawal *et al*[7], RSA factorization and CDH assumptions holds. We are going to consider the same adversary model with the purpose of justifying the claim above.

The attacks considered were; whether an attacker can infer sensitive information from observing the protocol messages, whether the misbehavior from one of the parties would allow the other to learn any other interest either than the one they have in common and lastly whether one of the parties could prevent the other from leaking an interest that the two of them have in common. The six attack scenarios can be categories into semi-honest or passive attacks and Malicious or Active attacks. Attack scenario 1 and 2 belong to the first while the remaining four attacks scenarios belong to the latter with attack scenarios 3 and 4 of non-common interest and attack scenarios 5 and 6 of common interest.

We consider an attack on messages sent by Alice, which can easily be implied to

Assumption: Decisional Diffie-Hellman hypopaper (DDH), $\langle S_i, f_e(S_i), S_j, f_e(S_j)\rangle$ for fixed values of i and j, with $f_e(x) = x^e$, is indistinguishable from $\langle S_i, f_e(S_i), S_j, Z\rangle$, when e is not given. Further, they conclude that for polynomial t and k, equation (5.1) is indistinguishable from equation (5.2)

$$\begin{pmatrix} S_1 & \dots & S_t & S_{t+1} & \dots & S_k \\ f_e(S_i) & \dots & f_e(S_t) & f_e(S_{t+1}) & \dots & f_e(S_k) \end{pmatrix} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (5.1)$$

$$\begin{pmatrix} S_1 & \dots & S_t & S_{t+1} & \dots & S_k \\ f_e(S_i) & \dots & f_e(S_i) & Z_{t+1} & \dots & Z_k \end{pmatrix} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (5.2)$$

These properties give us the following results:

Attack 1*. Bob cannot map S_i^a back to S_i if he does not know the value of a, which is known only to Alice.*

Proof: if Bob could map S_i^a back to S_i, Bob could compute $f_a^{-1}(Z)$, and by checking whether $f_a^{-1}(Z) = S_j$, or not, Bob can distinguish $\langle S_i, f_a(S_i), S_j, f_a(S_j)\rangle$ and $\langle S_i, f_a(S_i), S_j, Z\rangle$. This ensures that before Alice sends Bob the computed result in step 3, Bob is not able to learn any of Alice's interests. Carol observing S_i^a cannot learn S_i.

Attack 2*. Given the values of $S_1, f_a(S_1), \dots, S_t, f_a(S_t), S_{t+1}, f_a(S_{t+1}), \dots, S_k, f_a(S_k)$, Bob cannot compute the value of a.*

Proof: assume that Bob could compute the value of a, so he is able to compute $f_a(S_k)$, and by checking if $f_a(S_k) = Z_k$, Bob could distinguish equation (5.1) from equation (5.2)

$$\begin{pmatrix} S_1 & \dots & S_t & S_{t+1} & \dots & S_k \\ f_e(S_i) & \dots & f_e(S_t) & f_e(S_{t+1}) & \dots & f_e(S_k) \end{pmatrix} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (5.1)$$

$$\begin{pmatrix} S_1 & \dots & S_t & S_{t+1} & \dots & S_k \\ f_e(S_i) & \dots & f_e(S_i) & Z_{t+1} & \dots & Z_k \end{pmatrix} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (5.2)$$

This guarantees that Bob cannot obtain Alice's exponent, a. As a result, even if Bob knows the ID, V of an interest that he does not have, he cannot compute V^a. Therefore, he cannot detect if Alice has this interest by checking if V^a equals $S_i^a \ \forall i \in [m]$. The same applies to an eavesdropper observing the values.

In steps 6 - 8, Alice and Bob exchange their signed interest certificates for all their matching interests. These certificates are sent across a secured channel, so a passive eavesdropper, Carol, cannot learn Alice's or Bob's interests.

Attack 3*. Alice or Bob cannot learn non-common interest since a cheating partner is definitely detected by the other. Bob sending out $\{S_j^b, \dots, S_n^b\}$, and getting back this*

$\{S_1^a, \dots, S_m^a\}$ and $\{(S_1^b, (S_1^b)^a), \dots, (S_j^b, (S_j^b)^a)\}$ from Alice , Bob can learn only whether or not Alice has a number of elements in set $\{S_1, \dots, S_n\}$.

Proof: According to the above two properties, if and only if $(S_j^b)^a = (S_i^a)^b$, Bob can learn which S_i from when Alice matches one of her S_j in step 3. Otherwise, it is impossible for him to get the value of S_i. Namely, Alice could pair S_j^b with $(S_j^b)^a$ for $j \neq i$ in step 3, and Bob would think Alice has interest S_j instead of S_i if $(S_j^b)^a$ is in the intersection. Bob would send S_j to Alice. As a result, Alice is able to gain extra information without being detected. Exchanging the signatures of the signed interests will detect this attack. Because Alice receives messages from Bob that are symmetric to the messages that she sends to Bob, she is also unable to learn extra interests.

Attack 4*: Alice and Bob cannot use interests not signed by the IDS for each of them*.

Proof: Alice has to provide $sig_{Sk_A}(A\_ID \| S_i^a) \ \forall i \in [m]$ for all elements in her set to prove that she really is assigned this interest before the matching. Unlike the symmetric key-based based commutative functions, it is not possible to find $S^{1a^1} = S^a$ because of the way we create S and S^1 values and because of the discrete logarithm problem. This guarantees that Alice can use $(A\_ID \parallel S^a)$ only to prove her ownership of interest S but for nothing else.

Attack 5: *Alice and Bob cannot get useful information by replaying other users' responses.*

Proof: They have to run a signature-based authentication protocol before executing the matchmaking protocol. This prevents Alice and Bob from using signatures created by or assigned to other users.

Attack 6*: No one party can learn all the common interest.*

Proof: Let assume that Alice misbehave or tries to cheat. Alice wants only to explore Bob information but doesn't want to find a new friend.

Assume Alice and Bob have two interests in common. After Alice receives $(S_i^a)^b \forall i \in [m]$ from Bob in step 4, she knows that they have two common interests. Alice can lie and return the correct value of $(S_j^b)^a$ for only one of the two matched interests. In this case, Alice finds a new friend and learns more information than Bob does. However, this attack is not possible in our protocol due to the commitment in step 3 Alice has to execute this step honestly since she does not know which interests they have in common at this step. As a result,

she has to report $(S_j^b)^a$ and g^a and g^b correctly to Bob in step 3, since it is hard to find $h(x^1) = h(x)$ for $x^1 \neq x$. Otherwise, Bob will detect her malicious behavior immediately.

Also if Alice only wants to learn Bob's interests, but does not want to find a new friend. However, in step 3, she sends Bob a commitment to some random values, instead of a commitment to $(S_j^b)^a \forall\, j \in [n]$. If Alice does run step 3, but returns the random values to Bob, instead of $(S_j^b)^a \forall j \in [0, n]$, Bob is unable to detect this cheating since he cannot map $(S_j^b)^a$ back to S_j^b. Nevertheless, Alice takes the risk that Bob records the protocol messages. In this situation, Bob has the signed messages from Alice. In the same way, Bob can detect if Alice misbehaves in step 3. An alternative to recording the protocol messages probabilistically is using zero-knowledge proofs that prove that a $(S_j^b)^a$ was computed on the S_j^b revealed in step 2 and that the value of $\boldsymbol{a}$ used in this computation was identical to the value of $\boldsymbol{a}$ in S_i^a revealed in step 1 (without revealing a). But, these zero-knowledge proofs are expensive and would surge the computation (and communication) overhead, which is why we choose a probabilistic detection method instead.

Table 1 shows a comparison of our proposed protocol and two communicative encryption based on security against attacks scenarios.

Table 1: Anti-Attack Capability comparison

Protocols	Attack 1	Attack 2	Attack 3	Attack 4	Attack 5	Attack 6
Xie [8]	√	√	√	√	√	√
Agrawal's[7]	√	√	×	×	×	×
Our's	√	√	√	√	√	√

B. Performance Evaluation

This section will demonstrate the efficiency analysis of our scheme. By efficiency we mean that the proposed Protocol provides the desired function for privacy preservation during matchmaking in friend discovery among several users while incurring minimal computation, communication, infrastructure and overhead. We focused on the computation, communication and overhead used by our protocol and report its efficiency for privacy preservation in Mobile Social Networks.

Communication cost：TCP which stands for Transfer Control Protocol, is a connection-oriented Protocol which can ensure reliable transport but with a reliable transmission comes with a huge price that is the accuracy of the data content inspection must take up the computer processing time and network bandwidth. In particular, when TCP sends a paragraph, it starts a timer, waiting for the destination to confirm that it have receipt this message. If a timely feedback is not received in form of confirmation, it will resend the message. When TCP is received at the other end of the data from a TCP connection, it sends a confirmation. This confirmation is not immediately send, usually will be delayed a fraction of a second.

Computation cost: Public key algorithm, the security of RSA algorithm depends on the large sum Numbers decomposition, a public key and a private key is a function of two large prime Numbers. In order to ensure the RSA algorithm has enough encryption intensity, Electronic commerce the SET (Secure Electronic Transaction) agreement CA (Certificate Authority) to use 2048 bits RSA keys, Implementation of this paper is to use 1024Bit RSA key. The longer the length of the key,the greater the difficulty of the big sum decomposition, the greater the algorithm the higher safety it is. So, when RSA is to deal with data, a large number of module power operation is required, mode of power operation efficiency determines the data throughput of the RSA algorithm.

General mode of power operation using the binary method or k into the process.

Set Euler's Number binary representation for e

$$e = e_k - 1e_k - 2e_k - 3 \ldots e_0 = \sum_{i=0}^{k} e_i 2^i, e_i \in [0,1]$$

The k-base module and power operation process is as follows:

Input m, e, n: $output$: $m^e \, mod \, n$

Scanning r bit at a time, then, $k = 2^r$.

Calculating $m^w \, mod \, n$ in advance($w = 2$ and $4, \ldots, k-1$) and preserved.

The index into s are bit unit Fs, $i = 0, 1, 2, \ldots, s-1$).

To calculate M^{Fs-1};

For loop ($from \; I = s - 2 \; to \; I = 0$)

1. $c = C^2 mod\ n$;
2. $2. if\ F\ i \neq 0, then\ calculate\ C = C * M^{Fi} mod\ n; get\ result. That\ is\ c$;

$r = 1$, is the binary algorithm, it is the simplest form of k into the system. The algorithm's time complexity is $O(t) = (3/2)k^3$. The computation complexity of our protocol and that of Xie's is shown in Table 2. The Computational cost is evaluated using the number of Power Modular computation (PM), and the communicational cost is evaluated using number of messages transmitted.

Table 2: Comparison of Complexity

Protocols	Computation complexity	Communication complexity
Xie's	$2(N-1)(m+n)PM + 2(N-1)DH$	$(N-1)(m+n+5)$
Our's	$2(N-1)(m+n)PM + 2(N-1)DH$	$(N-1)(m+n+3)$

The computation complexity of our protocol is the same as that of Xie's however, the communication complexity in terms of is greatly reduced because the DH session key can be computed within matchmaking.

VI. CONCLUSION AND FUTURE WORK

Friend discovery in MSN makes use of a lots of infrastructure, architecture and protocols. The three key architectural design introduces different benefits and different levels of privacy preservation. The overhead of these architectures can be solved with efficient protocols. Central architecture makes the users totally dependent on the server which must be virtually online all the time. Distributed architecture allows users to manage their resources their own way whiles hybrid employs both the server providing management services whiles users determine their attributes for friend discovery certified by an authority.

We presented a work based on Xie's, proposing a new solution to the problem of privacy preservation in MSN. Our proposed system model is able to offer fast and efficiency friend discovery with reduced steps and overhead. We reduced the steps involved in the process. Setup procedures were minimized, signature-based authentication and revocation procedures. Our proposed protocol had reduced communicational complexity whilst maintaining the privacy of users. We attempted to find a solution to user anomaly in MSN which would offer users with total anonymity during matchmaking in friend discovery.

We presented a new method for friend discovery in mobile social network. We proposed a new distributed protocol based the work of Xie'seliminating the interest signer and PIS. In our protocol the trusted third party is not involved in the matchmaking. It is employed at the setup stage to determine credentials for matchmaking. Our major contribution is a reduced communicational cost with the same computational complexity and Security as that Xie's while maintaining privacy during friend discovery in MSN.

The implementation generated a time complexity of $O(t) = (3/2)k^3$of which the code was done in LINUX with C and implemented with OPENSSL and RSA scheme.

Although few challenges in the MSNs have been identified in this research, there are still many challenges that need to be addressed. Also, there are many opportunities for improving the effectiveness and efficiency of friend discovery in MSN. This section deliberates some of the future research possibilities by presenting them under future works. These include the followings:

1. A comprehensive security model will be designed for proposed protocol.
2. Real-time analysis will be done in the future to determine the real cost of implementation.
3. Google API to determine the location of the users for additional matchmaking attribute will be considered in details and implemented.
4. Ring signature will be replaced with RSA with aim of achieving user anonymity.

REFERENCES

[1] http://mobiforge.com/research-analysis/global-mobile-statistics-2014-part-a-mobile-subscribers-handset-market-share-mobile-operators [Accessed on 2015/02/10]

[2] N. Eagle and A. Pentland. "Social Serendipity: Mobilizing Social Software". IEEE Pervasive Computing, 4(2):28–34, 2005.

[3] K. A. Li, T. Y. Sohn, S. Huang, and W. G. Griswold, "PeopleTones: a system for detection and notification of buddy proximity on mobile phones", in Mobisys'08, June 10-13,2008,

[4] M. Li, u, and W. Lou. FindU: Privacy-Preserving Personal Profile Matching in Mobile Social Networks. In Proc. of Infocom 2011.

[5] Z. Yang, B. Zhang, J. Dai, A. Champion, D. Xuan, and D. Li, "*Smalltalker: A distributed mobile system for social networking in physical proximity*," in IEEE ICDCS '10, June. 2010.

[6] A-K Pietil¨ainen, E. Oliver, J. LeBrun, G. Varghese, and C. Diot. "Mobiclique: *Middleware for Mobile Social Networking*" In Proc. of WOSN'09

[7] R. Agrawal, A. Evfimievski, and R. Srikant, "Information sharing across private databases," in SIGMOD '03. New York, NY, USA: ACM, pp. 86–97, 2003

[8] Q. Xie, U. Hengartner. "Privacy-Preserving Matchmaking for Mobile Social Networking Secure Against Malicious Users", In Proc. 9th Int'l. Conf. on Privacy, Security (PST), and Trust 2011
[9] R. Zhang, J. Zhang, Y. Zhang, *et al*., PrivacyPreserving Profile Matching for Proximity-based Mobile Social Networking. IEEE Journal on Selected Areas in Communications, 2012
[10] R. Li and C. Wu, "An unconditionally secure protocol for multi-party set intersection," in ACNS '07, pp. 226–236, 2007
[11] M. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In Eurocrypt, 2004.
[12] C. Hazay and Y. Lindell. Efficient protocols for set intersection and pattern matching with security against malicious and covert adversaries. In TCC, 2008.
[13] S. Jarecki and X. Liu. Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection. In TCC, 2009.
[14] E. De Cristofaro, J. Kim, and G. Tsudik. Linear-Complexity Private Set Intersection Protocols Secure in Malicious Model. In Asiacrypt, 2010.
[15] C. Hazay and K. Nissim. Efficient Set Operations in the Presence of Malicious Adversaries. In PKC, 2010.
[16] S. Jarecki and X. Liu. Fast secure computation of set intersection. In SCN'10, 2010.
[17] L. Kissner and D. Song, "Privacy-preserving set operations," in CRYPTO '05, LNCS. Springer, pp. 241–257, 2005
[18] G. Ateniese, E. D. Cristofaro, and G. Tsudik. (if) size matters: Sizehiding private set intersection. In Public Key Cryptography, pages 156–173, 2011.
[19] Q. Ye, H. Wang, and J. Pieprzyk, "Distributed private matching and set operations," in ISPEC'08, pp. 347–360, 2008
[20] S. Jarecki and X. Liu. Efficient Oblivious Pseudorandom Function with Applications to Adaptive OT and Secure Computation of Set Intersection. In TCC, 2009.
[21] Ronald L. Rivest, Adi Shamir and Y. Tauman, "How to leak a secret" in C. Boyd (Ed.): ASIACRYPT 2001, LNCS 2248, pp. 552-556, 2001
[22] Emmanuel Bresson, J. Stern and M. Szydlo, "Threshold Ring signatures and Applications to Ad-hoc Groups" in CRYPTO 2002, LNCS 2442, , pp. 466-480, 2002
[23] R. Rivest, A. Shamir, and L. Adleman. "A method for obtaining digital signaturesand public-key cryptosystems." in *Com. of the ACM*, 21(2):120–126, Feb. 1978.
[24] A. Bender, J. Katz and R. Morselli, "Ring Signatures: Stronger definition and construction without Random Oracles.
[25] W. Dong, V. Dave, L. Qiu and Y. Zhang, "Secure friend discovery in Mobile Social Networks", in NSF Grants CNS-09161606, CNS-0546755 and CNS-0916309
[26] J. Ruiter and M. Warnier, "Privacy Regulations for Cloud Computing: Compliance and Implementation in Theory and Practice", Computers, Privacy and Data Protection: An Element of Choice, 2011 – Springer
[27] J. Kemp and F. Reynolds, "Mobile social networking: Two great tastes ",in Proc. W3C Workshop Future of Social Netw., Jan. 2009
[28] N. Kayastha, D. Niyato, P. Wang and E. Hossain, "applications, architectures, and protocol design issues for mobile social networks: a survey"
[29] Zhong, L. Bi, Z. Feng, and N. Li, "Research on the design of mobile social network services" in Proc. Int. Conf. Inf. Manage. Innovat. Manage. Ind. Eng., vol.2, pp. 458-46, dec. 2008
[30] P. Dhakan and R. Menezes, "the role of social structures in mobile ad-hoc networks" in Proc. ACM Southeast Regional Conf., , pp.59-64Mar. 2005
[31] www. en.wikipedia.org/wiki [Accessed on 2015/02/10]
[32] Nigel Smart, Crytography: An introduction (3rd Edition)
[33] Ronal Cramer, Victor Shoup, " a practical public key cryptosystem provably secure against adaptive chosen ciphertext attack"
[34] Set Theory available [online] at http://en.wikipedia.org/wiki/Intersection_(set_theory), [Accessed on 2015/02/10]
[35] B. R. Karki, A. Hamalainen, and J. Porras, "Social networking on mobile environment", in Proc. ACM/IFIP/USENIX Middleware Conf. Companion, pp. 93–94, 2008
[36] A. Gupta, A. Kalra, D. Boston, and C. Borcea, "Mobisoc: A middleware for mobile social computing applications", Mobile Netw. Appl., vol. 14, no. 1, pp. 35–52, Feb. 2009
[37] D. Brooker, T. Carey, and I. Warren, "Middleware for social Networking on Mobile Devices", in Proc. 21st Softw. Eng. Conf., Apr. 6–9, pp. 202–211, 2010
[38] A. Karam and Nader Mohamed, "Middleware for Mobile Social Networks: surve" IEEE, pp 1482 - 1490, 2012
[39] Zhu, Haojin, *et al*. "Fairness-aware and Privacy-Preserving Friend Matching Protocol in Mobile Social Networks." 1-1.
[40] C. Boldrini, M. Conti and A. Passarella "Contentplace: social-aware data dissemination in opportunistic networks" in Proc. ACM Int. Symp. Model. Anal. Simul. Wireless Mobile Syst., pp.203-210, Oct 2008
[41] PatientsLIkeMe. [online]. Available: http://www.patientlikeme.com/[Accessed on 2015/02/10]
[42] M. Netter, S. Herbst and G. Pernul, "Analyzing in Social networks – An interdisciplinary approach" IEEE, pp.1327-1334, 2011
[43] http://www.cesg.gov.uk/publications/index.htm#nsecret. [Accessed on 2015/02/10]
[44] N. Vastardis, and K. Yang, "Mobile Social Networks: Architectures, social properties, and key research challenges" IEEE, pp.1355 – 1371, 2012
[45] N. Kayastha, D. Niyato, P. Wang and E. Hossain, "Applications, Architectures, and protocol design issues for mobile social networks: A survey", in vol. 99, No.12 December 2011|proceedings of the IEEE. Pp.2130 – 2158, 2011
[46] D. Boneh, "The Decision Diffie-Hellman Problem",
[47] L. Law, A. Menezes, M. Qu, J. Solinas and S. Vanstone, "An Efficient Protocol for Authenicated key Agreement", in technical report CORR 98-05, Dept of C&O, University of Waterloo, Canada, March 1998.
[48] Y. Sang, H. Shen, and N. Xiong, "Efficient protocols for privacy preserving matching against distributed datasets", in ICICS '06. Springer-Verlag, pp. 210–227, 2006.
[49] Y. Sang, and H. Shen, "privacy preserving set intersection protocol secure against malicious behaviors", IEEE, pp.461-468,2007
[50] D. Dachman-Soled, T. Malkin, M. Raykova, and M. Yung, "Efficient robust private set intersection," in ACNS '09, pp. 125–142, 2009
[51] G. Ateniese, E. De Cristofaro, and G. Tsudik. "(If) Size Matters: Size- Hiding Private Set Intersection", In Proc. of PKC'11, pages 156–173, 2011
[52] S. Jarecki and X. Liu. "Fast Secure Computation of Set Intersection" In Proc. of SCN 2010, pages 418–435, 2010
[53] Surrogate Architecture Specification, Rev. 1.0, 2001. [Online]. Available: https://surrogate.dev.java.net/doc/sa.pdf, [Accessed on 2015/02/10]