Trustworthy Service Discovery for Mobile Social Network in Proximity

Chii Chang
Faculty of Information Technology
Monash University, Australia
Email: chii.chang@monash.edu

Satish Srirama
Institute of Computer Science
University of Tartu, Estonia
Email: srirama@ut.ee

Sea Ling
Faculty of Information Technology
Monash University, Australia
Email: chris.ling@monash.edu

Abstract—Mobile Social Network in Proximity (MSNP) is a new form of social network in which users are capable of interacting with their surroundings via their mobile devices in public mobile peer-to-peer (MP2P) environments. However, establishing such an MSNP faces several trust issues. A classic MP2P trust scheme usually requires high amounts of data transaction in order to identify the trustworthiness of service/content providers. This paper proposes a lightweight trustworthy service discovery scheme for service-oriented MSNP. The evaluation results show that the proposed schemes can reduce the overall transaction cost and are equally reliable to the basic schemes which require a large number of reputation rating data.

I. INTRODUCTION

Mobile Social Network in Proximity (MSNP) [1], [2], [3] is a MP2P-based social network in public ubiquitous computing environment. MSNP participants can either share content data directly from their Mobile Web Service (MWS) [4] or they can simply conduct their MWS to provide links that redirect content requesters to retrieve content from the providers’ Social Network Service (SNS) spaces.

Since MSNP operates in a public MP2P environment, the communication between MSNP participants involves trust issues. For example, a content provider’s content may not be consistent with its description metadata, or the service provided by a participant may exhibit malicious behavior. Since a reliable central management party for supporting trustworthiness is not available in MSNP, the environment requires a decentralised trust solution letting each MSNP participant manage the access control by itself. Performing trust management control in MSNP also faces challenge in latency because a stable third party entity to determine the trustworthiness is not available in MSNP. A requester who intends to determine the trustworthiness of a stranger’s service needs to refer to other participants’ past experience with the stranger’s services. Intuitively, mobile participants may have synchronised their trust-related data to their backend cloud storages so that these data can be retrieved indirectly and will not be affected by their movement. However, for the requester who needs to collect and process those trust-related data, his/her overall discovery performance will be affected and will result in high latency.

This paper presents a scheme to improve the speed of trustworthy service discovery in service-oriented MSNP by reducing transaction overhead and not relying on message forwarding in order to avoid the issues caused by unstable connectivity and resource constraint.
Definition 4: Service Provider Ratings—SPR. SPR = \{ID_k, Rates_k\} : 1 ≤ k ≤ |N| where: ID_k denotes the identification of SP_k. Rates_k = \{service^k, rate^k\} : 1 ≤ l ≤ |N| \} is a list of rating values of SP_k’s services. service^k denotes one of the SP_k’s services. rate^k denotes the rating value of service^k.

Definition 5: Recommended References—RR. RR = \{SType_m, ID_m\} : 1 ≤ m ≤ |N| where: SType denotes a semantic service type. ID_m = \{id^m, rate^m\} : 1 ≤ o ≤ |N| \} denotes a list of MSNP participants’ IDs that are recommended as the rating reference for SType_m services.

Definition 6: Reputation Rating Data—RD. Each MSNP participant has a RD file in its device local storage as well as its cloud storage synchronously. An RD file contains two sets of data—SPR and RR.

Listing 1. Simplified RD example

```xml
<key>Service Provider Rating</key>
<value>
  <key>SPID</key>
  <value/>
  <key>URI</key>
  <value/>
  <key>type</key>
  <value>semantic type value</value>
  <key>Rate</key>
  <value>rating value</value>
  <key>transaction records</key>...
</value>
</key>Recommended References...
<key>Semantic Service Type</key>
<value>
  <key>ID</key>
  <value>URL of RD</value>
  <key>other IDs...</key>
</value>
<key>other types...</key>
</value>
```

Listing 1 illustrates a simplified RD in hash map format. An RD file can be obtained from either friends or other proximal MSNP participants. The prerequisite condition is how the requester agent retrieves the RD from the other agents (either from friends or public proximal participants). In a generic Mobile Ad Hoc Network environment, it is commonly assumed that the requester agent will collect the RD by broadcasting or multicasting its request message to the other participants’ agents. This is not always applicable in MSNP. Fundamentally, MSNP operates in a dynamic public MP2P environment in which participants may not always be available.

To resolve the basic data retrieval problem in MSNP, each MSNP participant can utilise one or multiple backend public accessible cloud storage services to provide its RD to the others. The URL of the RD can be simply described in SDM. Hence, while the requester agent retrieves Service Description Metadata (SDM; e.g., WSDL) in the first phase of service discovery process, it can already identify where to retrieve the reputation rating data provided by the other proximal participants. As for the friends’ RD, since the requester has close connection with them, the requester would have already replicated their SDM files. Therefore, the requester agent always knows where to retrieve the RD of the requester’s friends.

One aspect in MP2P trust that was not addressed in most existing works is how service providers actively participate in the trustworthy service discovery processes. In real world services, providers always attempt to encourage consumers to use their services by using various schemes such as showing customers’ rating and reviews of their products and services. Although in an MP2P trust system, service providers should not hold the rating of their own services [5], they can still provide a list of previous interacting service consumers.

When a requester intends to retrieve a service provider’s reputation rating, the service provider can provide a PSC list. The requester can use the cid of PSC list to collect RD instead of collecting all the RD of friends or proximal strangers. This approach can reduce unnecessary data transmission. Moreover, MSNP agents can identify that a service provider who does not provide the PSC list can potentially be a malicious node unless the service provider is new to the MSNP. If an MSNP participant is new, it may not have any interaction record with any other participants either as a service consumer or as a service provider.

Considering the situation when a dishonoured service provider may provide an incomplete PSC list, which only describes a list of good records, the requester agent should not refer to the service provider’s PSC list to identify the service provider’s trustworthiness when: (1) In the case of recommendation from friends; If none of the cid found in PSC belongs to the requester’s trusted friends, the PSC should not be used. (2) In the case of recommendation from public; If none of the cid found in the PSC belongs to highly creditable strangers, the PSC should not be used.

The following sections describe the proposed scheme for trustworthy service discovery in service-oriented MSNP.

A. Selecting Recommenders Based on Friends and FOAF

Due to privacy issues, the information about a person’s trust rating value to his/her friends may not be accessible to other friends. However, the person can still provide a list of friends as RR for a particular service type. The friends’ Identifications (IDs) assigned in RR denote that the owner of RR trusts this list of participants’ judgement for a particular service type based on their past experience.

RR is generated and updated when an MSNP agent performs service by referring to the RD of its user. RR only contains the IDs of trusted friends for a particular service type. If a friend in this list has given a high rating to a bad service provider, the friend’s ID will be removed from the list. When a service provider ID is blocked, the friends who gave a good rate to the service provider will be removed from the corresponding RR. On the other hand, when the list is empty and the recommendation was from random picked friend, if a friend’s recommended service provider gives satisfactory recommendation to the requester, the friend’s ID would be added to the list.

There are two approaches to assign friends to RR:

(1) Based on experience. Since an RD provides a list of ratings, an agent is capable of identifying which friend of its user has the highest service interaction experience with a specific service type.
(2) Based on similarity. A user can assign their friends to
RR based on how similar their past rating to a particular
service type, i.e., using Pearson Product-moment Correlation
Coefficient.

Both approaches require a fair number of friends’ RD
replicated previously. For example, a user can replicate their
friends’ RD at home, then their agents can apply the ap-
proaches to identify RR before the user using MSNP applica-
tion outside.

The following algorithm outlines the steps for a requester to
identify the trust score of a service/content provider’s service
s ∈ S.

**Algorithm 1:**

**Step 1.** Identify a list of friends who have experience with
service—s. Requester retrieves PSC of the provider of s
(PSC_s). We expect that the requester has a list of friends’
IDs (denoted by FID, where FID = \{id_j : 1 ≤ j ≤ N\})
stored in the local memory of the mobile device. By searching
the intersection between all the cid in PSC_s and FID,
requester can find a list of friends who have service invocation
experience with s — MFID, where MFID = FID ∩ CID. If
|MFID| = 0 then the process goes to Step 3. Otherwise, continue
with Step 2.

**Step 2.** Identify matched recommended references. As
described previously, each MSNP participant has a RD. Let
MRR = \{rr ∈ RR : SType_{er} = SType_s\}, where SType_s
is the semantic service type of s that the requester intends to
invoke. RR is a list of friends’ IDs that are recommended for
identifying the reputation of a type of SType_s. Let
Rr1ID = MFID ∩ MRR. From Rr1ID, the requester agent
can identify the recommended friend(s) for SType_s that also
have experience with s, and refer the friend’s rating to s. If
|Rr1ID| = 0, the process goes to Step 3.

**Step 3:** Referring recommendation from recommended friend’s
FOAF. When the requester’s direct friends do not have experi-
ence with s, the requester will refer to the reputation rating
from FOAF.

Identify a friend with the highest experience as a recom-
nender and then based on the recommender’s RD to find the
friend of the recommender who has the highest experience with
SType_s, and who also has rated s. Once the FOAF is found,
the requester will refer to the FOAF’s rating of s. However, if
none of the FOAF has experience with s then the process will
proceed to the scheme described in Section II B—Selecting
recommenders based on public.

**B. Selecting Recommenders based on the Public**

In this section, we describe the scheme to identify a service
provider SP’s reputation score based on the public proximal
MSNP participants’ ratings.

**Definition 7:** Credibility — Cr. An MSNP participant’s Cr,
which is rated by the other peers, represents its reputation
as a recommender for a type of service. The more MSNP
participants’ IDs shows up in the RR of every peer’s RDs,
the higher the MSNP participant’s credibility is for being a
recommender of the corresponding service type.

**Algorithm 2:**

**Step 1:** Generating a candidate recommender list. While the
requester performs the service discovery process to find service
providers who can provide the service of interest, the requester
is also retrieving the RD of each proximal MSNP agent. This
step consists of the following two tasks:

1.1. Let PRRD be the set of RDs retrieved from all proximal
agents. PRRD = \{prrd_i : 1 ≤ i ≤ N\} where prrd_i denotes
the RD of each agent p_i. For each prrd_i ∈ PRRD, the
requester agent can identify that whether a p_i has interaction
experience with service provider s or not.

1.2. Let MPR denotes the matched PRRD in which MPR =
\{prrd_j ∈ PRRD|ID ∈ SPR_j = ID_s\}. ID_s denotes the ID
of service provider s. If ID_s is found in one of prrd_j’s SPR
but not in the PSC list of the provider of s, then either the
prrd_j is dishonoured or the provider of s is dishonoured.

Since the aim of this scheme is to identify the trust of s’s
provider, the final result will show its reputation score. How-
ever, dishonoured rating from the other participants will affect
the accuracy of the scheme. Hence, the requester agent has
to identify a recommender’s trustworthiness before referring
its reputation rating. Step 2 describes the process to identify a
recommender’s trustworthiness based on credibility.

**Step 2:** Identify the credibility of a candidate recommender.
A proximal MSNP participant’s credibility is computed based
on the other proximal MSNP participant’s rating. Suppose we
want to compute a proximal MSNP participant p_i’s credibility,
we will use PRRD excluding the RD of p_i. We use CRRD
to represent such a set of data, where CRRD = \{crrd_m : 1 ≤
m ≤ N\}. Step 2 consists of following two tasks:

2.1. Let Crp be the credibility of p, Crp = |\{crrd ∈
CRRD|ID_{crd} ∈ ID_p\}|, where ID_{crd} denotes an
MSNP participant’s ID in the RR of crrd, and ID_p denotes
p_i’s ID in MSNP.

2.2. Once the credibility of each PRRD’s owner p_i is com-
puted, the process goes to the next step.

**Step 3:** Identify the experience of a candidate recommender.
People trust a person who has more experience about a specific
subject. In existing works such as TEMPR [6], the experience
of p is directly related to the number of successful interactions
completed between p and the service provider. Here, we
consider the experience based on the type of service instead of
a particular service provider’s service. Because in the real
world, a person may not use a service the second time when
he/she had a bad experience with the service the first time.
However, the person may have a lot of of experience using the
same type of service provided by many different providers.
Hence, the person’s opinion is still valuable. For example, the
review of a senior computer machine reviewer, who has over
100 reviews of notebook computers from different brands, is
often being considered as more trustworthy than a junior reviewer
who has only reviewed less than 10 number of notebook
computers. Based on this assumption, the experience of p in
our model is based on p_i’s experience to a particular service
type. This step involves the task below:

Let STypeExp_{pi} → s be p_i’s experience to SType_s. The
experience value of \( p_i \) to \( STypeEx \) is computed by:
\[
STypeEx_{p_i \rightarrow s} = \sum_{p_i \in \sum_{p_j \in p} STypeEx_{p_j \rightarrow s}} \nabla \sum_{p_i \in p} \nabla
\]
where \( IR_{p_i}^{RD} \) is the interaction records of \( p_i \), in which \( IR_{p_i}^{RD} = \{ \nabla \nabla i \nabla \leq \nabla \nabla N \} \). \( STypeEx_{p_i \rightarrow s} \) denotes the service type of the invoked service recorded in \( ir_{i \rightarrow p} \).

**Step 4: Compute the trust score of a candidate recommender.**
The trust score of an MSNP participant is the average of its normalised credibility value and its normalised experience value. The normalised value is computed based on the overall comparison from all the other participants in \( P \). This step involves the following task:

For a particular MSNP participant—\( \varphi \in P \)—as a recommender of a service type \( Tr \), the trust score \( Tr_{\varphi} \) of \( \varphi \) is computed by the formula:
\[
Tr_{\varphi} = \text{avg} \left( \frac{C_{r_{\varphi}} + STypeEx_{r_{\varphi} \rightarrow s}}{\sum_{p_i \in p} C_{r_{p_i}} + \sum_{p_i \in p} STypeEx_{r_{p_i} \rightarrow s}} \right)
\]
where \( C_{r_{\varphi}} \) is \( \varphi \)'s credibility value. \( \sum_{p_i \in p} C_{r_{p_i}} \) denotes the sum of credibility values of all \( p_i \). \( \sum_{p_i \in p} STypeEx_{r_{p_i} \rightarrow s} \) denotes the experience of \( \varphi \) for \( STypeEx_s \). \( \sum_{p_i \in p} STypeEx_{r_{p_i} \rightarrow s} \) denotes the sum of all \( p_i \)'s experience for \( STypeEx_s \).

Based on the computation result, the requester can choose a number of MSNP participants that have the highest \( Tr_{\varphi} \) value to be its recommender to compute the reputation score of \( s \).

**III. Evaluation**

The evaluation consisted of two parts corresponding to the two schemes described in Section II-A and II-B.

We describe our evaluation approach below:

1. For each user record of a trust rating dataset, we considered the user as a requester in MSNP who had a set of trust rating records (denoted by R-set) which corresponds to the RD.

2. From the R-set, we separated the records into two subsets: rating of friends and rating of non-friends.

3. From the rating of non-friends subset, we used the proposed schemes to predict what was the requester’s rating for each rating of non-friends.

4. We also used the basic schemes (i.e., by simply referring to the ratings from all the rating of friends or all the friends of the corresponding users of rating of friends) to predict what was the requester’s rating for each rating of non-friends. Then we compared the results between the proposed schemes with the basic schemes.

5. Finally, we compared the data transaction costs between the proposed schemes with the basic schemes. We then applied a basic Cost-Performance Index (CPI) model to compare the schemes.

We have tested our proposed trustworthy service discovery scheme using the Advogato dataset of 26 May, 2013. The original dataset contains many records with empty rating values (Some users have not rated any other users). Since our proposed scheme requires a fair number of rating data to calculate the trust score of a person based on other users’ ratings, we have removed users who have less than 10 rating records from the original dataset. The original Advogato dataset does not specify the relationship between users (i.e., are they real friends or not?). However, from their trust ratings, we categoried the relationship of users into two groups: when two users rated each other as ‘Master’ level, they are ‘friends’. Otherwise, they are ‘non-friends’. The following sections present the evaluation cases and results.

**A. Selecting Recommender Based on Friends and FOAF**

The aim of this test is to show that the proposed scheme (described in Section II-A) requires less transaction cost but still can provide similar trust score measurement result as the basic schemes. The basic schemes use a simpler approach to determine a service/content provider’s trustworthiness based on the reputation rating of all the requester’s friends or all the requester’s FOAF. They are:

- **All Friends (AF).** The requester computes a service provider’s trust score based on the average rating values of all the requester’s friends who have rated the service provider.
- **All Friends of Friends as Recommended Reference (AFOAF).** The requester computes a service provider’s trust score based on the average rating value of all RRs of the requester’s friends. The RR in this scheme are simply the FOAF who have rated the service provider without additional filtering.

The proposed schemes are:

- **One High Experience Friend (HEF),** which corresponds to the description in Algorithm 1, Step 2.

- **One High Experienced FOAF (HEFHEF),** which corresponds to Algorithm 1, Step 4.

- **One Most Similar Friend (MSF),** which corresponds to Algorithm 1, Step 3.

In this test case, we firstly retrieved a list of user IDs (as requesters) from the dataset. Each user had a list of ratings consisting of the IDs of the persons who had been rated, and the corresponding rating level value. Our test focused on predicting the requester’s rating of each ‘non-friends’ (representing service providers who will be evaluated by the requester) based on ‘friends’ and ‘FOAF’.

We used the above five different schemes to perform the prediction to show that the proposed scheme is efficient approach to measure the trust score of a provider.

We assumed that the requester has replicated friends’ RD in local memory previously. Hence, at runtime, it can identify recommenders for computing the reputation score of a service provider without retrieving all friends’ RD directly from the friends’ MWS or their cloud storages. The replicated RD can only be utilised to identify recommenders. In order to find out the up-to-date reputation rating score from the recommenders, the requester still has to perform the request to retrieve the necessary RD directly from the friends’ MWS or their cloud storages. Depending on the scheme used, the required RD-retrieval process can be different.
Table I summaries the cases of different schemes that were used for testing and comparison. The Comparable Count (CCount) in the table represents the total number of rating records that have been used to test the scheme. Because each scheme relies on different criteria, the CCount differs. For example, not all the users have available friends or FOAF’s ratings to predict the trust rating of a specific user. Hence, such incomparable records have been excluded in the testing for that scheme.

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Comparable Count</th>
<th>Prediction Accuracy</th>
<th>AMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Basic</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AF</td>
<td>1010</td>
<td>0.633569</td>
<td>6</td>
</tr>
<tr>
<td>AFOAF</td>
<td>1075</td>
<td>0.642984</td>
<td>36</td>
</tr>
<tr>
<td>Proposed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HEF</td>
<td>1010</td>
<td>0.635335</td>
<td>1</td>
</tr>
<tr>
<td>HEFHEF</td>
<td>1010</td>
<td>0.640418</td>
<td>6</td>
</tr>
<tr>
<td>MSF</td>
<td>1010</td>
<td>0.579199</td>
<td>1</td>
</tr>
</tbody>
</table>

The values of ‘Average Minimum Transaction Required (AMT)’ in Table I were computed as follows:

AF scheme requires up-to-date reputation rating values from all friends. The average minimum transaction required is equal to the average number of friends of each requester under test, in which the average number of friends each recommender has is 6, which is the average number of ‘Master’ level ratings of each user in the Advogato dataset.

AFOAF scheme requires the highest transaction cost at runtime incurred by retrieving the up-to-date reputation rating values from all FOAFs. The total cost of the required transaction was the number of friends multiplied by the number of FOAF, which is 36.

In HEF scheme, since the requester has replicated the RD previously, the replicated old RD is sufficient for the requester to identify a HEF at runtime without consuming data transaction cost on retrieving new RD via the Internet. Once a HEF is found, the requester only needs to retrieve the up-to-date reputation rating value from the HEF. Hence, in this case, the transaction cost is 1.

HEFHEF scheme requires the minimum transaction values is 6, which is the sum of the transaction cost of retrieving RD from all friends of HEF.

MSF scheme incurs the same transaction cost as the HEF-based scheme.

In order to highlight the overall improvement of the proposed approaches (HEF, HEFHEF, MSF) compared to the basic approaches (AF, AFOAF), we have translated the results into a CPI model. Table II shows the CPI value of each approach. As the table shows, when direct friends are available as the recommenders of the reputation rating, the proposed HEF and MSF schemes provide better CPI values than the basic scheme—AF. When direct friends cannot be the recommenders, during which FOAF is needed, the proposed HEFHEF scheme gives a better CPI value than the general AFOAF scheme.

### B. Selecting Recommenders Based on the Public

The test described in this section corresponds to the scheme described in Section II-B that identifies which strangers’ reputation rating values are reliable based on the stranger’s experiences and credibility.

This test aims to show that the proposed scheme can improve the accuracy when the trustworthy service discovery process is based on public proximal MSNP participants’ rating scores. In this test case, we used the ‘non-friends’ of the Advogato dataset as the proximal strangers of the requester.

The test case compared the proposed scheme with the basic Naïve scheme. The two schemes are summarised below:

**Naïve Scheme.** The requester computes a service provider’s trust score based on the average rating values of all the requester’s ‘non-friends’ who have rated the service provider. The service provider is excluded from the list of ‘non-friends’.

**Proposed Scheme (Psch).** The requester computes a service provider’s trust score based on a selected recommender based on both credibility and experience computed from the ‘non-friends’ list. Same as the Naïve scheme, the service provider is excluded from the list of ‘non-friends’.

We also included two additional schemes—Experience Only (Exp Only) and Credibility Only (Credit Only)—in which the requester selects a recommender based on only experience and based on only credibility respectively. These two schemes were included because we wish to show that the proposed scheme (based on both credibility and experience) provides better prediction accuracy than the cases of only using one of them to predict the reputation rating value.

When referring to the ratings from the public, the average minimum transactions required were the same, because the requester had to collect all the proximal MSNP participants’ rating data in order to identify their credibility and experience. The value—7 is the average number of ‘non-friends’ that each user had in the Advogato dataset (See Table III).

In our test, we removed all the friends from the dataset. Each requester derived another user’s rating score based on other user’s rating values (i.e., public recommendations).

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Comparable Count</th>
<th>Prediction Accuracy</th>
<th>AMT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Psch</td>
<td>851</td>
<td>0.703078</td>
<td>7</td>
</tr>
<tr>
<td>Naïve Scheme</td>
<td>851</td>
<td>0.504942</td>
<td>7</td>
</tr>
<tr>
<td>Exp Only</td>
<td>851</td>
<td>0.686321</td>
<td>7</td>
</tr>
<tr>
<td>Credit Only</td>
<td>851</td>
<td>0.499681</td>
<td>7</td>
</tr>
</tbody>
</table>

**TABLE IV. Predictive Rating Accuracy Comparison of Different Schemes Based on Public**

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Naïve</td>
<td>0.5</td>
</tr>
<tr>
<td>Psch</td>
<td>0.7</td>
</tr>
<tr>
<td>Exp Only</td>
<td>0.69</td>
</tr>
<tr>
<td>Credit Only</td>
<td>0.5</td>
</tr>
</tbody>
</table>
Since the transaction cost of all schemes were the same, we did not need to calculate their CPI value to compare their performance in this case. As the result shows in Table IV, the accuracy of the Naïve scheme was 50%, which means that if the requester computes a provider’s trust based on the average trust rating scores from all the proximal MSNP participants, it will only have a 50% chance for the result to match what the requester expects. If the requester computes the provider’s trust score based on the most experienced MSNP participant’s rating (Exp Only), there is a 69% chance that the result will match what the requester expects. On the other hand, if the requester only refers to the trust score of the highest credible match what the requester expects. If the requester computes the provider’s trust rating scores from all the proximal MSNP participants, the requester had to collect all the rating data from all the proximal participants’ agents. Reducing the transaction cost in public-based trustworthy service discovery for MSNP requires further investigation. We consider this as one of our future research directions.

IV. RELATED WORKS

A number of works have been proposed to support trustworthiness in MP2P environments. While works proposed by [7] and [8] were focusing on how to improve the reliability of trust models by utilising the computation of a large number of trust-related data, resulting in insufficient processing speed in MP2P network [9], some authors [10], [11], [6] have proposed lightweight trustworthy service/peer discovery schemes for MP2P environments.

Reducing data transaction is a common strategy to improve the processing speed of trust in MP2P. [10] have proposed a group-based reputation scheme. Their design is based on super peer MP2P network, in which a super-peer (which is described as Power peer in their work) manages the reputation rating data of a group of mobile peers with similar movement speed. M-Trust [11] reduces reputation data transaction by selecting recommenders based on the confidence of the candidate recommenders. Similar to the fundamental strategy of M-Trust, TEMPR [6] also improves the trust processing speed by utilising the selective recommender approach. Distinguished from M-Trust, the TEMPR scheme computes direct peers’ (candidate recommenders who can directly interact with the requester) trustworthiness based on two scores: (1) the direct peers’ trustworthy rating from other unknown peers; and (2) the direct peers’ untrustworthy rating from other unknown peers.

Our work can be seen as an extension of TEMPR, designed specifically for service-oriented MSNP. The major difference is that we do not assume strangers’ application will always forward messages to assist other participants for the trust processes. Hence, a requester who intends to identify a provider’s trustworthiness has to obtain the reputation rating data by either directly invoking the data provider agent (if the agent provides the corresponding Web service operation) or by retrieving the data from the data owner’s cloud storage (based on the URL links described in the data owner’s SDM).

V. CONCLUSION

This paper presents a lightweight trustworthy service discovery scheme for service-oriented MSNP. The test results show that the proposed schemes can reduce the overall transaction cost and are equally reliable to the basic schemes which require large number of reputation rating data. Furthermore, although the proposed scheme for predicting the reputation of the service/content provider based on public proximal MSNP participants does not reduce the transaction cost, it can improve the chance of finding the reliable recommenders for retrieving the reputation ratings of content/service providers.

For future work, we plan to extend the current scheme and develop an adaptive lightweight trustworthy service discovery solution for a pure public MSNP using Mobile Cloud Computing technologies.

REFERENCES