DECENTRALIZED SELF-MANAGEMENT OF TRUST FOR MOBILE AD HOC SOCIAL NETWORKS

Juan Li and Qingrui Li

Computer Science Department, North Dakota State University, Fargo, USA
(j.li, qingrui.li)@ndsu.edu

ABSTRACT

The construction of social networks over mobile devices in the events or at locations has emerged as a new network paradigm. These new mobile social networks enable people to communicate and share their experiences without the need to have Internet access and with minimum required infrastructure. However, the functionality and security of such networks would be potentially undermined without an effective trust management scheme. Although many trust management systems have been proposed, few of them can be applied to these new mobile social networks because of the unique network and communication characteristics. This paper presents a novel trust management system, termed MobileTrust, to establish decentralized, secure and reliable trust relationships between mobile ad hoc social network participants. Specifically, the construction of trust models encompasses both scenarios that users are experienced with the network and users are unacquainted with the environment. The trust models cover various important factors of trust relationships in social networks, including the similarity of user profile, reputation, and history of friendship. A set of simulations is conducted to evaluate our system deployed in a mobile social network in the presence of dishonest users.

KEYWORDS

Mobile wireless network; Social network; Trust; Privacy

1. INTRODUCTION

Social networking has had a great impact on the communication of people all over the world. Simultaneously to the surge of social networking, mobile devices, such as laptops, PDAs, and cellular phones, have been widely used. A natural trend is to integrate social networking with mobile devices leading to a type of new applications – mobile social networks. There have appeared many of such applications. For example, MySpace and Facebook have provided limited versions of their services on mobile phones. Users of these sites interested in accessing the social networking applications can use their mobile devices while on the go. In this kind of mobile social networks, the mobile social network sites (the servers) are treated as a central authority with which the user can trust. The trust between users is based on pre-established social relationships, such as work colleagues, family members, and friends.

In practice, there exists tremendous need of building social networks spontaneously from mobile devices anytime, anywhere, even without the Internet or server infrastructure. Such social networks are particularly useful in the scenarios where Internet infrastructure is unavailable, ineffective, or expensive (for example, driving on a highway, traveling on a train, cruise, or plane). This motivates a new generation of social networks, mobile ad hoc social networks. The construction of such social networks over mobile devices in the events or at locations, such as conference, campus, shopping mall, and restaurants enable people to communicate and share their experiences without the need to have Internet access and with minimum required infrastructure. This kind of social network can also play important roles in special applications such as transportation planning that uses travelers' real-time travel information in disaster management, traffic congestion, parking slot availability, ride share

DOI : 10.5121/ijcnc.2011.3601

opportunities, and commercial advertisement systems that recommend reviews and coupons to the travelers. A few infrastructures, for example, Jambo Networks [17], Nokia Sensor [18] and MobiLuck [34], have been emerged as trials to build ad hoc social networks spontaneously from the mobile devices.

These new mobile social networks shift from the existing social networking archetype towards a mobile ad hoc networking mode that can potentially connect any type of devices that are equipped with short-range communication medium, such as Bluetooth and Wi-Fi. Unlike traditional social networks in which social communities are pre-built from the offline reality, users holding mobile devices will be automatically connected in the network, based on their profiles, contexts such as locations, and social behaviors. Because users of such mobile social network do not have any previous interactions, it is more important to establish an acceptable level of trust relationships among participating users. As specified by Dwyer et al., "Trust is a critical determinant of sharing information and developing new relationships. Trust is also important for successful online interactions" [5].

However, trust management is much more challenging in spontaneous mobile social networks than in traditional centralized environment because of the absence of central authority and network infrastructure, coupled with the dynamic nature of the network topology. No single user has a complete global view of another user's trust information; instead, information about user interaction is spread across the whole network. Collecting trust information or evidence to evaluate a particular user's trustworthiness is difficult due to the large scale of the network and the mobility of the users. The dynamic nature of users results in uncertainty and incompleteness of the trust. Furthermore, malicious users might tamper with trust information while it is stored locally. Resource constraints further confine the trust evacuation process to only local information, so that trust establishment would be based on incomplete and incorrect information.

The objective of this paper is to propose a new trust scheme, MobiTrust, for spontaneous mobile social network. The proposed MobiTrust system is fully decentralized and self-managed. It effectively addresses the aforementioned challenges and efficiently establishes trust relationships among participating users. In particular, in MobiTrust, first, we propose a comprehensive trust model which encompasses all the important factors special for spontaneous mobile social network. Second, we propose novel approaches to calculate trust in both scenarios that users are familiar with the network environment and users are new to the network with little prior participation experience. Third, we propose an effective scheme to collect and propagate trust information in the network for future reference and verification. Our simulation experiments demonstrate the effectiveness of our proposed model.

Our previous work [26] initialized the research on trust management in spontaneous mobile social networks. In this paper, we extend it significantly by adding three novel approaches to assign trust to users in the mobile network (Section 2). In particular, we consider three typical scenarios: (1) users have sufficient knowledge of their activity context to construct trust setting rules, (2) users do not have sufficient knowledge of the context to generate explicit trust policies but they can collect sufficient useful information from the network, and (3) users neither have much knowledge of the environment nor can gather enough available information from the network in a short period. In the first scenario, we apply clustering methods to the local applications of the mobile device, and summarize the trust configuration of the applications in each cluster as context-based policy rules. In the second scenario, we propose collaborative filtering techniques to predict trust values from users' trust assignment history. In the third scenario, we present a dynamic model, which adjusts the weights of three key factors [26] of the trustworthiness in mobile social network, to compute the trust values.

The rest of the paper is organized as follows. Section 2 details the trust model, MobiTrust. In Section 3, we evaluate the proposed method and show the effectiveness of MobiTrust with a comprehensive set of simulations. Related work and concluding remarks are provided in Sections 4 and 5, respectively.

2. TRUST COMPUTATION

We adopt the definition of trust proposed by Golbeck [1], in which user *A* trusts user *B* if *A* commits to an action based on a belief that *B*'s future actions will lead to a good outcome. To compute trust in a spontaneous mobile social network, the first step is to facilitate the integration of trust into the network. That is to have a computation of trust that captures the social features while being narrow enough to function in the environment of a spontaneous mobile social network. When computing trust, we consider three typical categories of scenarios that can cover almost all situations in practice: (1) users have sufficient knowledge of their activity context to construct trust setting rules, (2) users do not have sufficient knowledge of the context to generate explicit trust policies but they can collect sufficient useful information from the network, and (3) users neither have much knowledge of the environment nor can gather enough available information from the network in a short period. Next, we discuss the details of how to compute trust for these three scenarios.

2.1. Experienced Users

In practice, users may be familiar with the network as they may have participated in the network before or they have prior knowledge of the network. For these users, they have specific expectations of the type of users, data, and applications that will be in the network. Based on these expectations and/or their prior experiences of the network, the trust policy rules can be generated either through user configuration or by the system automatically.

Note that users with prior experience or knowledge of their environment can sufficiently exploit the information of the context. Context awareness has been studied for about two decades, and the existing technologies have enabled a mobile device to discover, collect, and take advantage of contextual information such as location, time, nearby people or devices, and user activity [27]. Amongst the various methods of context categorizations, an effective approach is to categorize context into *user-centric context* and *environmental context* [28]. Examples of *user-centric contexts* are user's dynamic behaviors (such as her scheduled tasks and current activity) and user's profile data (such as her interest, habit, preference, working area, home area, etc.); while *environmental contexts* may include physical environment (such as time, location, noise, light, etc.) and social environment (like surrounding people, traffic jam, discount information, etc.). As we can see, from the point of view of environmental contexts, there may exist a huge number of scenarios of dynamic mobile environments and mobile applications. However, if we concentrate on the user-centric contexts, the number of possible scenarios for any given mobile user is always finite and limited. Therefore, we focus on the user-centric contexts. In this scenario, a clustering method can be applied to individual mobile users to categorize the contexts with various trust policy rules. For example, users' current activities are an important contextual attribute which affects or even dominates the computation of trust values. Since in many cases, users' activities are tightly coupled with the available applications on their mobile devices, we cluster the mobile applications according to their categories or semantics similarity. One method to compute the semantics similarity is presented in Section 2.2.

There are various kinds of clustering methods in the literature. Unfortunately, the state-of-art heuristic methods such as k-Means [29] and k-Medoids clustering are not suitable for our problem because it is difficult for the users or the system to determine the value of *k* beforehand. Therefore, we adapt a bottom-up hierarchical clustering method called AGNES [30]. By clustering, the applications on mobile devices will be categorized; each application category

will be associated with a trust policy rule. New applications can either be grouped into existing categories or create a new category.

We use some concrete examples to elaborate the clustering approach. Assume user *A* has mobile applications "My Conference" and "My Class Forum" on her mobile device. These applications may have different functionalities in their respective conference or class contexts, but both are categorized under the same cluster called "Meeting Cluster". The Meeting Cluster will be attached with a trust policy rule defined as following: all users at the same location in a specified time interval will be assigned a high trust level T_h. Another example shows a more complex context: in the "Mall Forum" application, users who join the social networks constructed from the users in the same shopping mall will assign higher trust values to those who have visited the same stores at the same day, so that they can share their shopping experience and recommendations.

2.2. Inexperienced Users with Sufficient Hints from Networks

A unique characteristic of mobile ad hoc social networking is that it involves spontaneous users; it is common that users never participate in these networks before and lack knowledge of the network environment. Here the knowledge of the network includes the information of other possible users, the details of data content, the types of the applications, etc. Given so much uncertainty, it is challenging for users to pre-define trust policy rules. However, users may be able to predict their unknown trust assignments based on the available trust assignments in the network. The prediction is based on the reasoning that users of similar profiles are likely to assign the same trust values to the same target. To achieve the goal of accurate trust prediction, we apply collaborative filtering techniques.

Collaborative filtering [31] is a kind of knowledge discovery techniques that is popularly used in recommender systems. The recommender systems help users find the items they would like to purchase at e-commerce sites by producing a list of recommended items for a given user. The basic idea of collaborative filtering algorithms is to predict users' ratings on un-rated items based on the opinions of other like-minded users. We adapt it to our trust computation problem.

Formally, in mobile ad hoc social networks, each user u has a $m \times m$ matrix M, where m is the number of users in the network. The cell $M[i, j]$ records the trust value that user i assigns to user j. If user i has not assigned the trust to user j yet, $M[i, j]$ is equal to 0. We assume that user u collects the trust assignments by other users who are willing to reveal them (similar to releasing ratings in the recommender systems), and records these trust values in M. Our goal is to make use of these trust assignments from the network to predict user u's trust to any user, if it is not assigned yet, as accurate as possible.

The basic idea of our solution is to find top-k users who have similar profiles as user u and who have assigned trust values to user v. The trust that u will assign to v will be computed as

$$trust(u, v) = \frac{\sum_{\forall u' \in N, trust(u', v) \neq 0} \boldsymbol{sim}(\boldsymbol{u'}, \boldsymbol{u}) * \boldsymbol{trust}(\boldsymbol{u'}, \boldsymbol{v})}{\boldsymbol{m}},$$

where m is the number of users who have assigned trust to user v. Intuitively, the trust setting is calculated based on how the similar users have assigned the trust to the same user.

The foundation of this scheme is a metric that measures users' profile similarity. Various vector similarity measurement metrics (e.g., cosine similarity [32], Pearson correlation [33]) have been proposed to measure the distance of vectors. However, most of them only consider the exact match; very few of them take semantics of the items in the vectors into consideration. As

semantics plays important roles in social computing, we extend the previously reported distance-based approaches [19], [20], [21] to accurately measuring the semantic similarity between user profiles. Our proposed approach extends the previous approaches by supporting multiple ontologies and improves the accuracy by integrating additional factors, such as the depth of a node in the ontology hierarchy and the type of links.

Definition 1 *(Keywords Distance).* Assume that the profile of user u can be represented as a vector of keywords $P_u=\{C_1, C_2, \ldots C_n\}$. The semantic distance between two concepts C_a and C_b is defined as:

$$dis(C_a, C_b) = \frac{1}{2}\left(\frac{\sum_{i\in path(C_a to C_p)} w_i dis(C_i, C_{i+1})}{\sum_{i\in path(C_a \text{ to } C_{root})} w_i dis(C_i, C_{i+1})} + \frac{\sum_{j\in path(C_b to C_p)} w_j dis(C_j, C_{j+1})}{\sum_{j\in path(C_b \text{ to } C_{root})} w_j dis(C_j, C_{j+1})}\right),$$

where C_p is the common ancestor of C_a and C_b in the hierarchical ontology graph, C_{root} is the root of the tree, C_{i+1} is C_i's parent, and w_i is the weight of edge presented as a distance factor.

Definition 2 *(Concept Similarity).* The concept similarity between two concepts C_a and C_b is defined as:

$$sim(C_a, C_b) = 1 - dis(C_a, C_b).$$

Definition 3 *(Profile Similarity).* Given two profiles P_x and P_y, the similarity between the two profiles is defined as:

$$sim(P_x, P_y) = \frac{\sum_1^n \max_{j\in[1,m]} sim(Cx_i, Cy_j)}{n},$$

where n is the number of concepts in profile P_x and m is the number of concepts in P_y. If $sim(P_x,P_y)$ is larger than a user-defined similarity threshold t ($0<t\leq 1$), the profile P_x is said to be semantically related to P_y.

The similarity measure defined above efficiently integrates the edge weight and the depth information. The semantic distance between two concepts is the sum of their distance to their common ancestor. To integrate the depth factor, the distance is normalized by the distance to the root. In this way, nodes at lower layers receive a higher similarity score.

An issue in the profile similarity evaluation is privacy. Because profiles may contain users' private information, some users may not be willing to reveal their private profile to others, especially strangers. Then how to measure the similarity of two users without revealing their private profiles is an important issue in this scenario. To address this issue, we design a privacy-preserving scheme to measure the similarity of user profiles. In this scheme, users' profiles are encrypted. We adapt the private set intersection protocol [22, 23], in which, two or more parties, each having a private dataset, can compute the intersection of their sets without revealing to each other any of the remaining elements. For example, suppose that party A has set $\{a_1, a_2, a_3, a_4\}$ and party B has set $\{a_1, a_2, b_1, b_2\}$. Then both A and B can learn that $\{a_1, a_2\}$ is the

intersection set. However, *A* cannot learn that *B* has b_1, and b_2, similarly *B* cannot learn that *A* has a_3 and a_4.

Several cryptographic solutions have been proposed recently for the privacy-preserving set intersection problem. We adopt the protocol [22] based on the use of homomorphic encryption and balanced hashing. We assume the user profiles are composed of a set of keywords. By applying the private set intersection protocol in [22] on the sets of keywords, the intersection between the user profiles that correspond to specific matching interests is returned. The complexity of the protocol is $O(m \cdot n)$, where *m* and *n* are the number of private triples in two input profiles. Obviously, the protocol is secure because no user learns more than the computed intersections of their private profiles.

2.3. Inexperienced Users without Sufficient Hints from Networks

Collaborative filtering techniques will be effective if user *u* can collect sufficient details of similar users who have assigned trust values to user *v*. However, it is possible that user u fails to do so, especially when she is new to the network and has not gathered much information from the network yet. In this case, we define a trust model that can quantify user *u*'s trust of user *v*. In particular, we define the computation model in *Definition 4*, which includes all important functional properties of trust in this environment.

Definition 4. Assume *A* and *B* are two users in the mobile social network. The trust value of *A* to *B* is defined as:

$$Trust(A,B) = \alpha \times Sim\big(Prof(B), Prof(A)\big) + \beta \times Rep(A,B) + \gamma \times fof(A,B)$$

in which:

$$0 \le Sim\big(Prof(B), Prof(A)\big) \le 1$$
$$0 \le Rep(A,B) \le 1$$
$$0 \le fof(A,B) \le 1$$
$$0 \le \alpha, \beta, \gamma \le 1$$
$$\alpha + \beta + \gamma = 1$$

In the above definition, $\mathrm{Sim}\big(\mathrm{Prof(B)}, \mathrm{Prof(A)}\big)$ evaluates the similarity between two user profiles, $\mathrm{Prof(B)}$and$\mathrm{Prof(A)}$. Function $\mathrm{Rep(A,B)}$ returns the reputation value of *B* from *A*'s point of view. Function $\mathrm{fof(A,B)}$ presents the common "friends" both *A* and *B* have contacted before. α, β, and γ are parameters that provide for differences in focus on the different components.

From the definition we can see that the defined trust of spontaneous mobile social network has following properties:

- The defined trust is asymmetric, i.e., "how much *A* trusts *B*" may give a different answer than "how much *B* trusts *A*". Employing such an asymmetric measurement reflects human judgment.

- The defined trust is personal. In the above definition, the trust value of *B* also includes affecting factor of *A*. This means trust is inherently a personal opinion. Different users may evaluate trustworthiness about the same person differently.

- The defined trust is not perfectly transitive. The definition of trust supports the idea of transitivity. Assumes we have another user *C*, and we have the fact that *A* highly trusts *B*, and *B* highly trusts *C*. Through the definition, it is highly possible that A trusts *C*, but it does guarantee that *A* will highly trust C.

This definition of trust encompasses all of the most important social factors in a spontaneous mobile social network. Due to the unique characteristic of spontaneous mobile social network and the inherent unreliability of the wireless medium, many of the functional properties of trust cannot be easily obtained. For instance, without a central server, we may not know the history between people's interaction and a particular user's reputation in general. Thus, a very important task of trust computation is to collect these trust factors from the network. In the rest of the test, we elaborate each of the major components of the trust model and present the strategy of extracting and propagating these trust factors.

2.3.1. Trust factor based on user profile similarity

In social network, people tend to trust others with similar interest or experiences. As shown in [2], there was a strong and significant correlation between trust and similarity; the more similar are the two persons, the greater is the trust between them. When there is no other trust evidence, for instance at the initial stage of the social network, this can be effectively used as a trust measurement. To measure the similarity between users, we compare their profiles. Profiles include personal information and usually include the users' opinions and ratings of items. This information can be used to compute how much one user should trust another. To protect users' privacy, profile information can be encrypted. Users periodically publish their (encrypted) profiles to their immediate neighbors. Other users evaluate the similarity between their profiles with a particular user by asking their neighbors. We use the similarity measurement methods proposed in Section 2.2.

2.3.2. Trust factor based on reputation

Reputation is the opinion or a social evaluation of the public towards an entity based on past experiences. We distinguish two types of reputation, personal reputation and global reputation. The personal reputation is recorded directly from a user's observation. Each user will also propagate this information so that the global reputation can be updated based on the accumulated personal reputation. Therefore, we define reputation as:

Definition 5. Assume *A* and *B* are two users in the mobile social network. *B*'s reputation value from the point of *A* is defined as:

$$Rep(A,B) = \theta \times per\_rep(A,B) + \lambda \times glob\_rep(B)$$

in which:

$$0 \le per\_rep(A,B) \le 1$$
$$0 \le glob\_rep(B) \le 1$$
$$0 \le \theta, \lambda \le 1$$
$$\theta + \lambda = 1$$
$$\theta = \frac{c}{c+n}, \qquad \lambda = \frac{n}{c+n}$$

In the above definition, $\mathrm{per\_rep(A,B)}$ is *A*'s personal observation of *B*'s reputation. $\mathrm{glob\_rep(B)}$ represents *B*'s global reputation. θ and λ are parameters that provide for differences in focus on the different components. Note that *B*'s popularity affects the calculation of θ and λ, the larger the value of *n* is, the larger the parameter λ is. Here *c* is a configurable constant which is used to manipulate and balance the weights of *A*'s personal observation and *B*'s global reputation and popularity, the value of *c* can be tuned according to various practical conditions.

Definition 6. A peer *B*'s global reputation $glob\_rep(B)$ is defined as:

$$glob\_rep(B) = \frac{\sum_{i=1}^{n} per\_rep(P_i, B)}{n}$$

in which: *n* is the number of users who rated user *B*. P_i is a particular user that once rated *B* before. The global reputation of *B* is defined as the average of the personal reputations *A* has received so far.

User *A*'s personal observation of user *B* can be easily found (if they interacted before) from the history information stored at *A*'s local memory or disk. Managing global reputation of *B*, however, is a tough task. It involves problems, such as where to store the global reputation? How to update it? How to extract it? In a mobile social network, there is no server to store reputations for users. Rather reputation values have to be stored in a decentralized manner. To avoid collusions and blackmailing, we distribute every user's public reputation in multiple nodes.

We assume each user is identified by a public/private key pair. After their interaction/transaction, user *A* can rate/comment user *B*, and vice versa. Besides storing *B*'s rating locally, *A* also gossips this rating together with its user ID and signature to the network. When a user *C* receives multiple ratings of user *B*, *C* will merge these ratings according to the rater's ID.

Before making friends with user *B*, user *A* needs to verify the reputation of *B*. In order to do that, *A* broadcasts a reputation query with a Time to Live (TTL). All users having *B*'s reputation stored will reply the reputation information of *B* to *A*. As mentioned, the global reputation of a user is based on the accumulated ratings collected.

2.3.3. Trust factor based on history of "friends"

The trust factor based on history of "friends" utilizes the transitive property of trust. Although trust is not perfectly transitive, "there is, however, a notion that trust can be passed between people." [4] If two users share common friends, these friends can bridge the trust gap between them. Assume two users, *A*, and *B*, successfully constructed their friendship. Each of them would sign the other's ID with his (her) signature, and exchange their certificates. Users will keep the singed document and certificate locally for future use. When two strangers, say *A* and *C*, find they once had a common friend *B*, they can trust each other in some degree based on their trust to *B*. To verify that *B* is the common friend, *A* and *C* will use their stored certificate to verify the signature. This way, the system does not need to maintain the keys for participants.

Definition 7. Assume *A* and *B* are two users. friends(A) represents all of the users who once were *A*'s friends. Similarly, friends(*B*) represents a group of *B*'s friends. The transitive trust of *A* to *B* based on the common friends they once had is defined as:

$$fof(A,B) = \frac{|\text{friends(A)} \cap \text{friends(B)}|}{|\text{friends(A)}|}$$

In the above equations, "∩" denotes set intersection, while "‖" represents set cardinality. The more friends they share, the more they can trust each other.

2.3.4. Weights of trust factors

Definition 4 assigns weights α, β, *and* γ to three trust factors. In order to get an accurate result of the trust value, it is essential to decide good values for α, β, *and* γ. Since our trust computation model presented above is a dynamic model and is typically applied to a process in which a mobile user learns about a new environment gradually, we propose our dynamic adaptation methods to quantify the values of α, β, *and* γ. The computation formulas are listed as follows.

$$\beta_{ca}(A,B) = \frac{\text{Number of participants who provide reputation ratings for B}}{\text{Number of current participants in network}}$$

$$\gamma_{ca}(A,B) = \frac{\text{friend(A)} \cup \text{friend (B)}}{\text{Number of current participants in network}}$$

$$w_{\beta} = \frac{\beta_{ca}(A,B)}{\beta_{ca}(A,B) + \gamma_{ca}(A,B)}$$

$$w_{\gamma} = \frac{\gamma_{ca}(A,B)}{\beta_{ca}(A,B) + \gamma_{ca}(A,B)}$$

$$\beta(A,B) = w_{\beta} * \beta_{ca}(A,B)$$

$$\gamma(A,B) = w_{\gamma} * \gamma_{ca}(A,B)$$

$$\alpha(A,B) = 1 - \beta(A,B) - \gamma(A,B)$$

Intuitively, β and γ are computed from β_{ca} and γ_{ca}, two parameters that truly behave the weights of the trust factors of reputation and friendships. It is not difficult to prove that the sum of β and γ is no greater than 1, as β and γ are computed in propotion to β_{ca} and γ_{ca}, both of which are no greater than 1.

The defined weights α, β, and γ are asymmetric; *A* has high β weight assigned to *B* does not mean vice versa. This fits the natural asymmetric property of trust.

We must note that α, β, and γ values are not fixed; they can be adapted as more accurate when users collect more information from the network. When a new mobile user *A* joins the network, she might have few friends and receives few reputation ratings initially. At this stage, the weights β and γ on her reputation and friendship are low, while the weight α on her profile similarity is high. This means that user *A* relies more on profile similarity than the other two trust factors at the moment, because there are not enough data of the other two factors that can be used for trust computation with confidence. As user *A* spends more time in the network, she

may make more friends and receive more reputation ratings. As consequence, the computation of trust on A will have higher weights (i.e., β and γ values) on reputation and friendship. In this case, profile similarity will be less important (i.e., take lower weights as smaller α values) than in the initial phase. This respects the natural interaction of social relationships and trust in the network.

3. EVALUATION

We conducted a set of experiments to evaluate our proposed trust model, MobiTrust. We present and discuss the results of the experiments in this section.

For the first scenario in which experienced users are deployed to the mobile social network in their familiar environments, trust assignments are based on the rules users have predefined for the applications. It is obvious that the predefined rules can accurately express users' trust policy as illustrated by the example in Section 2.1. We could not evaluate the accuracy of trust rules defined by the user; therefore, we did not perform further experiments to evaluate the performance of this scenario.

Due to the lack of real large-scale mobile social network data, for the second scenario of inexperienced users with sufficient hints from the network, we collected raw data from a population of real online social network users. In particular, we collected detailed trust/privacy information from a group of Facebook users. The privacy preferences for different friends reflect Facebook users' trust towards these friends. We built a Facebook application [35], which allowed us to collect privacy/trust preferences data from our surveyed objects, i.e., Facebook users. We collected data from 59 participants. We use part of the data as training data and the rest of the data as testing data to evaluate our collaborative filtering-based trust recommendation strategy proposed at Section 2.2.

Figure 1 shows the accuracy of our proposed recommendation strategies verses the number of friends used in the training stage. The results are promising: by using a small part of friends' information, our recommender can achieve a good average accuracy.

Figure 2 compares our semantics-enhanced trust recommender with the semantics-free trust recommender. As expected, the semantic-enhanced approach outperforms the semantic-free approach, because adding semantics can better capture subtle semantic similarity between users.

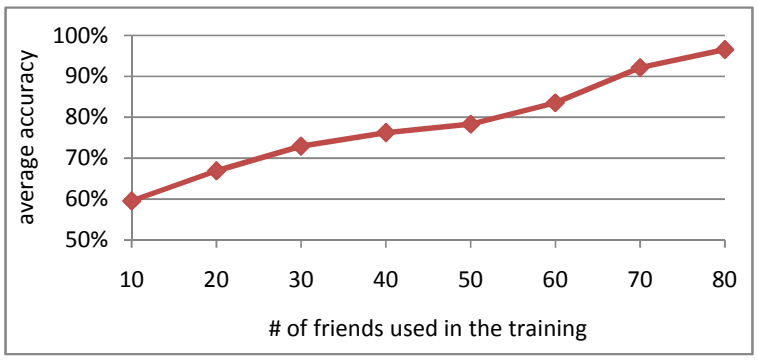

Figure 1: Average accuracy of the proposed trust recommendation strategy

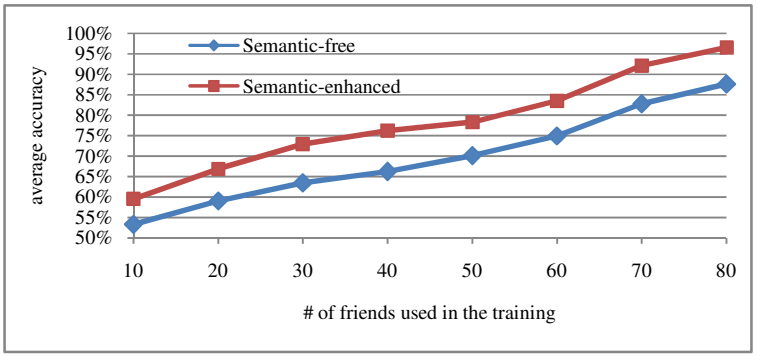

Figure 2: Comparison of semantic-free and semantics-enhanced recommendation approaches

A set of simulations are performed to evaluate the scenario of inexperienced users in an environment without sufficient hints. An enclosed ad hoc network environment was considered. The enclosed area that contained different nodes was off an area of 200 m x 200 m. The density of the nodes was adjusted throughout the simulations. The mobility of the nodes was similar to that of the "random waypoint" model as reported in [24]. In the random waypoint model, initially, the nodes are randomly distributed within the enclosed area. Each node has a randomly picked destination, towards which, the node moves at a predetermined speed. Once a node reaches its destination, the node pauses for a predefined interval of time, and then it repeats this movement pattern. The transmission range of a node was predetermined to be 10 m.

In the simulated mobile social network, nodes/users provide services to each other. Each node has its own set of generated profile. We assign 100 types of services randomly distributed across nodes of the enclosed area. There are two types of nodes in the network: honest nodes and dishonest nodes. Honest nodes provide the services they claim they have. We assume that node only provides services that match its profile, i.e., the semantics of the service profile is similar to the semantics of the profile of the node. The dishonest nodes claim that they have every service they are asked for, i.e., they reply with a bogus query-hit to every query they receive, without being able to provide the real requested services. Dishonest nodes are not always dishonest. They have a small amount of time of being "honest" in their life time. The various simulation parameters and their default values are listed in Table 1.

TABLE 1
PARAMETERS USED IN THE SIMULATIONS

parameter	**range (default)**
network size	200-2000 (1000)
environment area	200m*200m
node moving speed	1-20m/s (1m/s)
node transmission rage	10m
node pause time	0s-80s (20s)
query possibility per node per time slice	10%
TTL	4

no. of walkers	3
no. of keywords of user profile	1-10
type of services	100
services provided per node	1-5
node similarity threshold	0.6
% of bad nodes	0-50% (10%)
% of "good" behavior of bad nodes	10%-30% (10%)
α in Trust formulation	0-1 (0.2)
β in Trust formulation	0-1 (0.7)
γ in Trust formulation	0-0.3 (0.1)
Trust threshold	0.35-0.58 (0.5)

Figure 3 illustrates the performance of our proposed trust model by testifying their ability to recognize "bad" replies according to their trust knowledge. We tested different trust factors: (a) profile similarity only, (b) reputation only (c) combining of profile similarity, reputation and common friends. For comparison, we also show the result in (d) trust free situation. We keep the total number of nodes to 1000 and there are 10% percent of bad nodes. We did not test the trust factor of "common friends" separately, because this factor cannot provide enough trust evidence if using independently.

As shown in Figure 3, the proposed trust model and their individual trust factors dramatically improve the system performance by reducing the percentage of bad replies received. As time going, the system can build the reputation of participating nodes. Therefore, the performance of reputation factor improves as time increases. The performance of trust based on profile similarity does not change over time. Through evaluating the similarity between the profile of the service provider and the query, trust model based on profile similarity can detect the bogus replies. This factor is especially important at the initial stage of the social network, when users do not have other trust evidences. Note that in this experiment, the performance of this similarity factor is affected by our assumption: nodes only provide services that match its own profile interests. As expected, combing all three factors can achieve the best result. The ratio of these three factors is (2:7:1), i.e., α=0.2, β=0.7, and γ= 0.1.

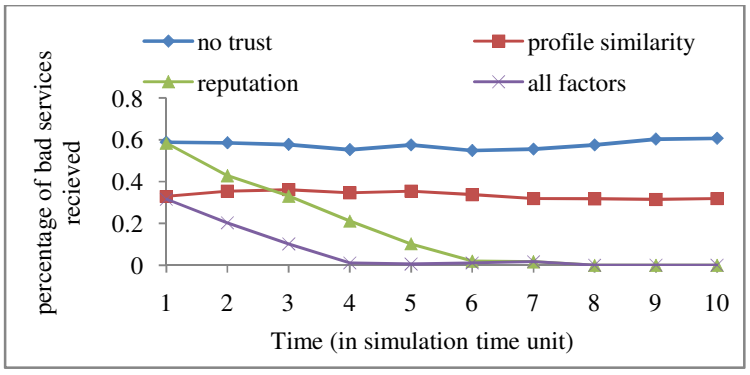

Figure 3: Performance of trust factors over time

We also observed the parameters of the trust model, α, β, and γ, also have an impact on the performance of the trust model. Figure 4 illustrates the trust model with different parameters. When α is larger, the system performs well even at the initial stage. When β is larger, the performance improves dramatically as the time going, and eventually performs better than the performance of models with smaller β. An application should set the values of α, β, and γ according to its properties, such as the typical life time of the network, the similarity between the service providers' profile sand their services, etc.

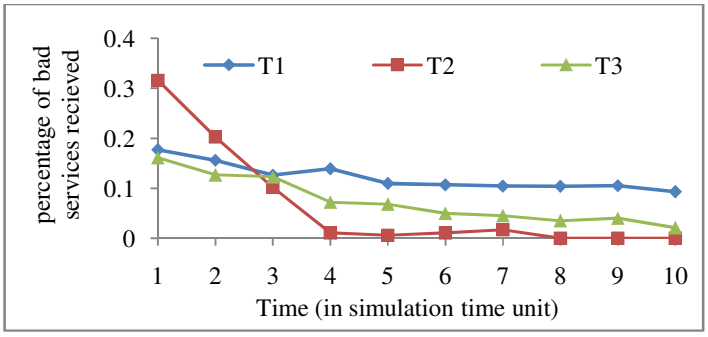

Figure 4: Performance of trust model with different parameters.
T_1: α=0.7, β=0.2, and γ= 0.1. T_2: α=0.2, β=0.7, and γ= 0.1. T_3: α=0.3, β=0.5, and γ= 0.2.

The trust model helps users to detect "bad" users. However, it may also misclassify "good" users that do not have high trust values as untrustworthy, especially when the trust threshold is selective. This is a common for all trust management system. Figure 5 and Figure 6 plot the false positive and false negative rate for query hits with various trust thresholds. Note the "false positive" rate here does not mean that the system goes wrong. It just demonstrates that some "honest" nodes may not be trusted by others because of their low trust value. A system should carefully pick the threshold to balance the tradeoff between false positive and false negative.

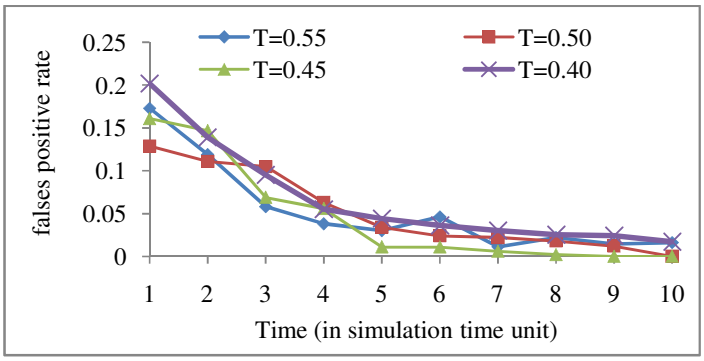

Figure 5: False positive rate over time for different trust thresholds.

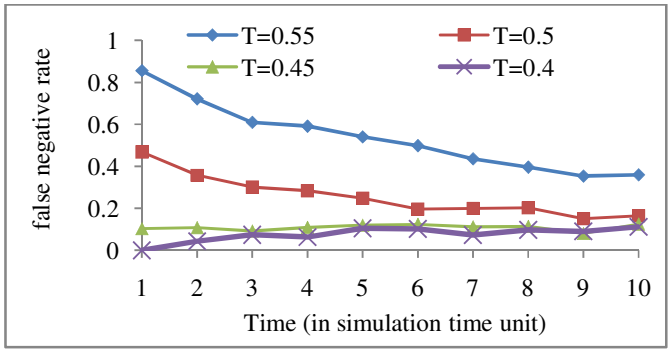

Figure 6: False negative rate over time for different trust thresholds.

To evaluate the performance of our trust model in a hostile environment, we varied the number of bad users in the network. Figure 7 shows the percentage of false matches in networks with different percentage of bad nodes. Without trust management, the rate of false matches is very high, even for relatively small percentage of bad nodes. By using the MobiTrust model, the rate of false matches is much less, even for the networks with many malicious nodes.

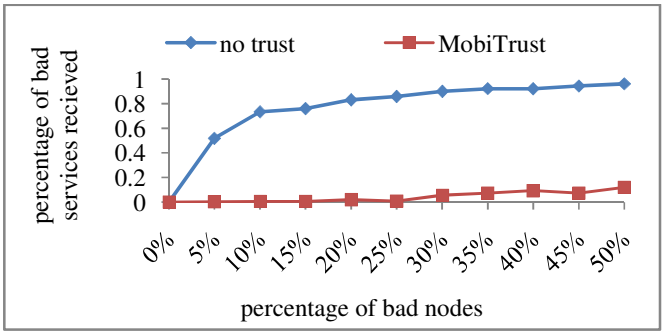

Figure 7: Performance of MobiTrust in networks with varied percentage of bad nodes.

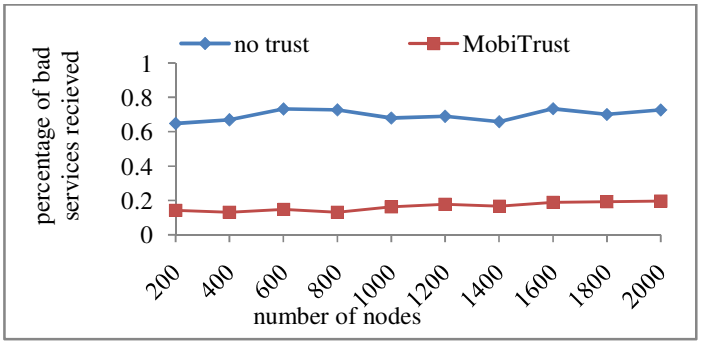

Figure 8: Performance of MobiTrust in networks with varied network size.

Figure 8 illustrates the performance of the trust model in network with varied number of nodes. It can be seen that MobiTrust performs well when the network size increases.

4. RELATED WORK

There is a large body of research studying trust in social networks. Golbeck proposed an algorithm, TidalTrust [15], for inferring trust relationships between people in social network.

TidalTrust uses a recursive search method to compute trust based on the social paths connecting people in the social network, and the trust ratings on those paths. In another work [25], the authors investigated features of profile similarity and how the profile similarity relate to the way users determine trust. They have shown that there is a correlation between users' profile similarity and their trust.

From its underlying network structure, spontaneous social network is a mobile ad hoc network (MANET). Managing trust in MANET has been studied in many works for different purposes such as secure routing [6, 7], authentication [10], intrusion detection [8, 9], and access control [11]. In our work, we address trust issue from a quite different prospective: construct secure and trustworthy social relationships between mobile ad hoc nodes. Therefore, the approaches we proposed are different from previous work.

Trust management has also been studies in other similar scenarios, such as peer-to-peer system. One of the most widely cited P2P-based trust algorithms is EigenTrust [13]. A peer maintains trust rating of other peers with which it has interacted. For one peer to determine the trustworthiness of another peer with which it has not interacted, it infers the trustworthiness based on the presence of pre-trusted peers. The EigenTrust algorithm calculates trust using a method similar to the PageRank algorithm [14] used by Google for rating the relevance of web pages to a search query.

Reputation is a fundamental concept in many situations that involve interaction between mutually distrusting parties. Damiani et al. [12] propose an overlay protocol to manage reputation for peer-to-peer networks, in which reliability of a resource can be established by distributed polling. In the COllaborative REputation mechanism (CORE) [16], reputation takes into account a task-specific functional reputation.

5. CONCLUSIONS

Recent years have witnessed a few social networking applications that use ad hoc communications rather than costly Internet access. In such a network, users must interact in highly dynamic and unpredictable environments, so the computational problem of trust, that is, determining how much one person in the network should believe in another person to whom they did not contact before, is extremely challenging. By exploring the special characteristics of mobile social networks, we designed a novel trust management model. In our trust model, we categorized the user's familiarities with a mobile social network into three typical scenarios, investigated the characteristics of each scenario, and proposed the corresponding approaches of trust computation for all scenarios. Contextual attributes, clustering method, and collaborative filtering techniques are applied either separately or in combination to different scenarios according to their respective properties. Detailed methods and formulas were proposed to either construct the policy rules or quantify the values of trust between users in a dynamic mobile social network. In addition to the theoretical modeling and analysis, the simulations testified that the proposed trust model can effectively evaluate users' trustworthiness and maintain satisfactorily performance with the varying network environment.

REFERENCES

[1] Jennifer Golbeck. 2005. Computing and Applying Trust in Web-based Social Networks. Ph.D. Dissertation.

[2] Ziegler, C.-N., & Golbeck, J. (2006). Investigating Correlations of Trust and Interest Similarity. *Decision Support Services*.

[3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[4] J. Golbeck and J. Hendler. "Inferring Trust Relationships in Web-based Social Networks". *ACM Transactions on Internet Technology*, Vol. 6, No.2, pp. 497-529, 2009

[5] Catherine Dwyer, Starr R. Hiltz, Katia Passerini, "Trust and privacy concern within social networking sites: A comparison of Facebook and MySpace", *the Thirteenth Americas Conference on Information Systems* , 2007

[6] Zouridaki, B. L. Mark, M. Hejmo and R. K. Thomas, "Quantitative Trust Establishment Framework for Reliable Data Packet Delivery in MANETs," *Proc. 3rd ACM Workshop on Security for Ad Hoc and Sensor Networks*, Alexandria, VA, Nov. 7, 2005, pp. 1-10.

[7] Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas, "Robust Cooperative Trust Establishment for MANETs," *Proc. 4th ACM Workshop on Security of Ad Hoc and Sensor Networks*, Alexandria, VA, 30 Oct. 2006, pp. 23-34.

[8] S. Buchegger and J.Y.L. Boudec, "A Robust Reputation System for P2P and Mobile Ad-hoc Networks," *Proc. 2nd Workshop on the Economics of Peer-to-Peer Systems*, 15 Nov. 2004.

[9] T. Ghosh, N. Pissinou, and K. Makki, "Towards Designing a Trust Routing Solution in Mobile Ad Hoc Networks," *Mobile Networks and Applications*, vol. 10, pp. 985-995, 2005.

[10] Weimerskirch and G. Thonet, "A *Distributed Light-Weight Authentication Model for Ad-hoc Networks," Proc. 4th Int'l Conf. on Information Security and Cryptology (ICISC 2001)*, 6-7 Dec. 2001.

[11] H. Luo, J. Kong, P. Zerfos, S. Lu, and L. Zhang, "URSA: Ubiquitous and Robust Access Control for Mobile Ad Hoc Networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 12, no. 6, Dec. 2004, pp. 1049-1063.

[12] E. Damiani, S. De Capitani di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. "A reputation-based approach for choosing reliable resources in peer-topeer networks." In *Proc. ACM Conference on Computer and Communications Security (CCS)*, pages 207–216. ACM, 2002.

[13] Kamvar, S. D., Schlosser, M. T., & Garcia-Molina, H. The eigentrust algorithm for reputation management in p2p networks. In *Proc. of the 12th International World Wide Web Conference*, 2004.

[14] Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). The pagerank citation ranking: Bringing order to the web. Technical Report 1998, Stanford University.

[15] Golbeck, J. (2005). Computing and Applying Trust in Web-based Social Networks. Ph.D. thesis, University of Maryland, College Park, MD, USA.

[16] Pietro Michiardi, Refik Molva "CORE: A Collaborative Reputation Mechanism to enforce node cooperation in Mobile Ad hoc Networks", *Sixth IFIP conference on security communications, and multimedia (CMS 2002)*, 107-121.

[17] Jambo Networks: http://www.jambo.net.

[18] Nokia Sensor: http://www.nokia.com/

[19] J. Li and S. Vuong, "SOON: A Scalable Self-Organized Overlay Network for Distributed Information Retrieval", in *19th IFIP/IEEE International Workshop on Distributed Systems (DSOM)*, 2008, pp.1-13.

[20] T. Pedersen, S. Patwardhan, J. Michelizzi, "WordNet: Similarity-Measuring the Relatedness of Concepts," in *19th National Conference on Artifical Intelligence (AAAI)*, 2004.

[21] R. Rada, H. Mili, E. Bicknell, M. Blettner. "Development and Application of a Metric on Semantic Nets," *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 19, no. 1, pp. 17-30, 1989.

[22] M. Freedman, K. Nissim, and B. Pinkas. Efficient private matching and set intersection. In *Advances in Cryptology – Eurocrypt '04*, volume 3027 of LNCS, pages 1–19. Springer-Verlag, May 2004.

[23] L. Kissner and D. Song. Private and threshold set-intersection. In Advances in Cryptology – *CRYPTO'05*, August 2005.

[24] C. Bettstetter and C. Wagner. "The spatial node distribution of the random waypoint mobility model". In *Proc. WMAN*, 2002.

[25] Jennifer Golbeck, "Trust and nuanced profile similarity in online social networks", *ACM Trans.* Web, Vol. 3, No. 4. (2009), pp. 1-33.

[26] J. Li, Z. Zhang, and W. Zhang;
"MobiTrust: Trust Management System in Mobile Social Computing"; *IEEE TSP'2010*: 3rd IEEE International Symposium on Trust, Security and Privacy for Emerging Applications

[27] Guanling Chen , David Kotz (2000). A Survey of Context-Aware Mobile Computing Research. No. TR2000-381

[28] Feng, Ling and Apers, Peter M.G. and Jonker, Willem (2004) Towards Context-Aware Data Management for Ambient Intelligence. In: *15th International Conference on Database and Expert Systems Applications*.

[29] Kanungo, T.; Mount, D. M.; Netanyahu, N. S.; Piatko, C. D.; Silverman, R.; Wu, A. Y. (2002)."An efficient k-means clustering algorithm: Analysis and implementation". *IEEE Trans. Pattern Analysis and Machine Intelligence* 24: 881–892.doi:10.1109/TPAMI.2002.1017616. Retrieved 2009-04-24.

[30] J. Han and M. Kamber. Data Mining: Concepts and Techniuqes. Morgan Kaufmann Publishers, San Francisco, CA, 2006.

[31] J. S. Breese, D. Heckerman, and C. Kadie. Empirical analysis of predictive algorithms for collaborative filtering. Technical Report MSR-TR-98-12, Microsoft Research, 1998.

[32] P.N. Tan, M. Steinbach & V. Kumar, "Introduction to Data Mining", Addison-Wesley (2005), ISBN 0-321-32136-7, chapter 8; page 500.

[33] J. L. Rodgers and W. A. Nicewander. Thirteen ways to look at the correlation coefficient. The American Statistician, 42(1):59–66, February 1988.

[34] MobiLuck, http://www.mobiluck.com/en/

[35] Facebook survey tool: http://apps.facebook.com/research_survey/