



## A practical reputation system for pervasive social chatting

Zheng Yan<sup>a,b,\*</sup>, Yu Chen<sup>c</sup>, Yue Shen<sup>b</sup>

<sup>a</sup> The State Key Laboratory of ISN, Xidian University, PO Box 119, No. 2 South Taibai Road, 710071 Xi'an, China

<sup>b</sup> Department of ComNet, Aalto University, Otakaari 5, 02150 Espoo, Finland

<sup>c</sup> Human-Computer Interaction Group, EPFL – I&C GR-PU, BC 145, Station 14, CH-1015, Lausanne, Switzerland

### ARTICLE INFO

#### Article history:

Received 23 February 2012

Received in revised form 18 September 2012

Accepted 8 November 2012

Available online 13 December 2012

#### Keywords:

Trust

Reputation system

Social networking

Trust/reputation visualization

Human-computer interaction

Pervasive social networking

### ABSTRACT

A Mobile Ad Hoc Network (MANET) is becoming a practical platform for pervasive social networking. For example, people chat with each other via MANET for instant social activities. How to help mobile users build up trust in pervasive social chatting is becoming an important and interesting issue. By applying a method for usable trust management, we designed PerChatRep, a reputation system for pervasive social chatting based on the result of a need assessment survey. We evaluated the effectiveness and robustness of PerChatRep through simulations. Furthermore, we implemented the system by applying Nokia N900 smart phones as MANET nodes based on a distributed energy-efficient social networking platform. We further conducted a two-session controlled user experiment to investigate the impacts of PerChatRep on mobile users. Results show the usefulness and user acceptance of PerChatRep.

© 2012 Elsevier Inc. All rights reserved.

### Contents

1. Introduction . . . . .	557
2. Background and related work . . . . .	557
3. Research method . . . . .	559
4. Need assessment . . . . .	560
4.1. Design . . . . .	560
4.2. Participants and results . . . . .	561
5. System design and implementation . . . . .	562
5.1. Definitions . . . . .	562
5.2. System structure . . . . .	562
5.3. Reputation generation . . . . .	563
5.4. Implementation . . . . .	565
6. Evaluation results . . . . .	566
6.1. Algorithm evaluation based on simulations . . . . .	566
6.2. System evaluation based on user experiment . . . . .	569
7. Conclusions and future work . . . . .	571
Acknowledgments . . . . .	571
References . . . . .	571

\* Corresponding author at: The State Key Laboratory of ISN, Xidian University, PO Box 119, No. 2 South Taibai Road, 710071 Xi'an, China.

E-mail addresses: zhengyan.pz@gmail.com, zyan@xidian.edu.cn, zheng.yan@aalto.fi (Z. Yan).

## 1. Introduction

A Mobile Ad Hoc Network (MANET) has a good prospect of becoming a practical platform for instant social activities [1]. MANET is a collection of autonomous nodes that communicate with each other by forming a multi-hop radio network and maintaining connectivity in a decentralized manner. A social group could be instantly formed based on MANET not only by people socially connected, but also those physically in proximity, such as groups for purchase, resource sharing and social events. For example, Groupon (<http://www.groupon.com>) provides daily digests of group purchase activities to users; eRideShare (<http://www.erideshare.com/>) helps people with similar driving routes to share car riding; Last.fm Festival (<http://www.last.fm/festivals>) suggests a list of music festivals to users near the event locations. A user could chat with strangers nearby for instant social needs. This kind of pervasive social chatting is an essential complement for Internet social networking, thus very valuable for mobile users, especially when Internet or mobile networks are temporarily unavailable or costly to access.

Trust plays an important role in the pervasive social chatting for reciprocal activities among nearby strangers. During the instant social activities, users are not necessarily acquaintances but more likely strangers. Therefore the users need to balance between the benefits received in such reciprocal activities and the risk of communications with strangers. In this context, it is important to figure out how much users should trust with each other in order to make a decision. This introduces a demand to provide a practical reputation system for MANET-based pervasive social chatting that could intelligently assist mobile users and encourage benevolent behaviors. On the other hand, the physical proximity among MANET nodes introduces an additional concern on privacy. How to provide identity/trust management with privacy enhancements is another practical issue needed to solve.

In this paper, we develop PerChatRep, a reputation system for pervasive social chatting by applying a methodology for usable trust management in order to provide good usage experience and make the designed system easily accepted by users [6]. We design a reputation scheme for PerChatRep to address the concern of users on trust during pervasive social chatting and their preferences on reputation visualization. We implement the system using Nokia N900 as MANET nodes based on a distributed energy-efficient MANET platform [1]. PerChatRep apply a centralized Trusted Server (TS) to provide accurate reputation when the system adopts pseudonyms for each node to enhance its privacy; solve inconsistent reputation problem in MANET, and protect the system from potential attacks [8,36]. Particularly, the local and general reputations are respectively evaluated by each individual node and the TS based on ephemeral and historical experiences. We validate the effectiveness of PerChatRep through both simulations and a two-session controlled user experiment with regard to its correctness, robustness, usefulness and user acceptance.

The rest of the paper is organized as follows. Section 2 reviews related work. We introduce the method applied in our research in Section 3. Following the steps of usable trust management development, a need assessment survey was conducted to investigate how users consider and expect to cope with a reputation system for pervasive social chatting in Section 4. Next, we describe the design and implementation of PerChatRep in Section 5. Section 6 reports simulation results that show the correctness and robustness of PerChatRep reputation generation, and conducts a user experiment to test the impact of introducing PerChatRep into pervasive social chatting. We further analyze the data collected from the user experiment to ascertain research results and implications. Finally, we conclude by discussing the contributions of this paper and suggesting future work in the last section.

## 2. Background and related work

Several research groups have focused on social activities based on mobile ad hoc networks. Stanford MobiSocial Group has developed Junction, a mobile ad hoc and multiparty platform for MANET applications [27]. Micro-blog [3], developed by SyNRG in Duke University, helps users to post micro-blogs tagged by locations. AdSocial [4], introduced by ETH Systems Group, provides a pervasive social communication platform. However, trust and reputation aspects in pervasive social networking are not considered in these projects. Traditional centralized social networking systems (e.g., facebook) have not taken user privacy into concern. They cannot satisfy instant social networking demands, especially when users do not have Internet connection, but with location proximity with each other.

In industry, quite a number of companies, such as Microsoft, Nokia and Intel have conducted research in the area of pervasive social networking (PSN). For example, Microsoft Research Asia has developed EZSetup system in order to let a mobile user find services provided by his/her neighbors [43]. The Nokia Instant Community developed by the Nokia Research Center provides an instant social networking platform to allow people in vicinity to communicate, get to know, and share information with each other [1,44]. Similarly, Intel Berkeley Lab runs a project named Familiar Stranger based on mobile devices to extend our feelings and relationships with strangers that we regularly observe but do not interact with in public places [7]. However, issues on trust management for security assurance and privacy enhancement need serious research in order to deploy a practical pervasive social networking system that can be easily accepted by mobile users.

Trust and reputation mechanisms have been widely studied in various fields of distributed systems, such as ad hoc networks, peer-to-peer (P2P) systems, grid computing, pervasive computing and e-commerce [2]. Trust is the belief of the reliability, integrity, ability, or character of an entity. Reputation is a measure derived from direct or indirect knowledge/experience on earlier interactions of entities and is used to assess the level of trust put into an entity [11]. Many

mechanisms have been developed for supporting trusted communications and collaboration among computing nodes [9–11]. Examples are FuzzyTrust system [16], the eBay user feedback system [17], PeerTrust model [18], an objective trust management framework (OTMF) for MANET [19] and Credence – a robust and decentralized system for evaluating the reputation of files in a P2P system [20]. Some work evaluates trust based on social relationships [12]. In these researches, trust can be modeled, calculated and thus expressed using a value. However, none of the above studies consider how to evaluate trust and reputation based on social networking behaviors and experiences, especially in the context of pervasive social chatting. None of them protect user privacy. Thus it is hard to directly apply them into PerChatRep. Some factors influencing trust explored from a user study like what we did for PerChatRep are never considered in the previous work. Moreover, only little work in the literature develops a reputation system driven by the concern of users [6].

A lot of work has been conducted regarding user interface (UI) design in order to improve user trust, mainly for web sites and in the context of e-commerce [5]. In many existing web services (e.g., eBay.com and Amazon.com), reputation values (mostly in a Likert scale) are displayed based on rating in order to assist user decision. However, due to little research in pervasive social networking, the study on interface design of MANET communication applications and services is insufficient [13]. In PerChatRep, we use a reputation icon to indicate the local reputation of each node during chatting and provide detailed information about the local reputation evaluated by the node and the general reputation issued by the trusted server. They are interface design elements that provide the cue of trust information in the pervasive social chatting. However, little previous research investigated the effects of visualizing reputation on mobile users in the context of PSN [13]. Prior art left room for further studies on reputation visualization, in particular, on how to provide trust information to mobile users. This is one of our research targets in PerChatRep.

Reputation system architecture is generally classified into two main types: centralized and distributed [21]. The system architecture determines how ratings and reputation scores are communicated between participants in a reputation system. In the literature, distributed trust evaluations have been studied in MANET, but seldom the solutions support node privacy [9,10,22]. This could cause potential attacks such as bad mouthing attack or unfair rating attack by artificially inflating or deflating the reputation of a specific node [8]. Most existing systems maintain a statistical representation of reputation by borrowing tools from the realms of game theory [23–25] and Bayesian analytics [26]. These systems try to counter selfish routing misbehavior of nodes by enforcing nodes to cooperate with each other and counter any arbitrary misbehavior of nodes. The concept of data centric trust in volatile environments, such as ad hoc networks, was introduced to evaluate the node trust based on the data reported by it [22]. However, little attention has been paid to the chatting reputation issue based on MANET with node privacy as a main concern. On the other hand, practical reputation systems generally apply a centralized server to collect feedback for reputation generation (e.g., eBay [17], Yahoo auctions [28]). However, many existing systems (e.g., Amazon and eBay) lack considerations on the credibility of user ratings. This greatly influences the quality of produced reputation. The usage of pseudonym and the ease of its change additionally complicate the picture by allowing participants to effectively erase their prior histories. PerChatRep adopts a hybrid reputation system architecture, where reputation is evaluated in a distributed way, but with the support of a centralized trusted server.

Sharing reputation information in the ad hoc networks introduce extra cost of communications. The purpose of reputation sharing is to make the reputation of a node known to all other nodes and decrease the detection time. Thus maintaining and disseminating indirect reputation information incur overhead at both the individual node and the network. OCEAN [30] discounts second-hand reputation exchange and only utilizes local reputation based on direct observations in order to achieve a reasonable performance. PerChatRep concerns both local and general reputations by aggregating local experiences and global experiences together. By deploying the trusted server, the overhead of reputation maintenance and dissemination is eliminated among MANET nodes.

Inconsistent reputation problem (i.e., different nodes may have different reputation values for the same node) often occurs in the ad hoc networks due to subjective reasons and/or different local experiences. This makes it hard to distinguish correct reputation ratings from reputation voting messages. LARS (Locally Aware Reputation System) was proposed to deal with selfish behaviors and malicious behaviors (e.g., packet dropping and unfair rating) [31]. In LARS, the reputation of a node is derived from direct observation and the exchange of second-hand reputation information is disallowed. In PerChatRep, we apply the trusted server to unify general reputation based on the local experience of the individual node. This general reputation value is issued to the node by the server. Serving as the initial value of reputation, it is further evolved based on the experience newly collected at the node to generate local reputation. In addition, the above process is periodically iterated. Thereby, we avoid the inconsistent reputation problem and eliminate user reputation inaccuracy caused by multi-hop reputation dissemination and the change of node pseudonyms since the trusted server holds the real identifier of each PerChatRep node. Reputation generation is based on first-hand experience and direct votes at both the TS and the node.

A reputation rating system based on past behavior of evaluators was proposed in [29]. Trust in the evaluator indexes its impact on the rating system. The trust value is dynamically adjusted based on past estimation performance. In PerChatRep, we apply an opinion deviation factor or the local reputation to discount the on-chat voting in the generation of local reputation. Applying the opinion deviation factor can overcome potential bad mouthing attack, as shown in our simulations.

Nowadays, reputation systems still face a number of potential attacks. Malicious users may artificially inflate or deflate reputations [5,17,28,32]. They could cooperate together to impact the reputation of a specific user. The system could be vulnerable to such attacks as on-off attack, independent/collaborative bad mouthing attack, and conflict behavior attack, ballot stuffing attack, and newcomer attack [33,34]. The usage of pseudonyms introduces new challenges since it makes it

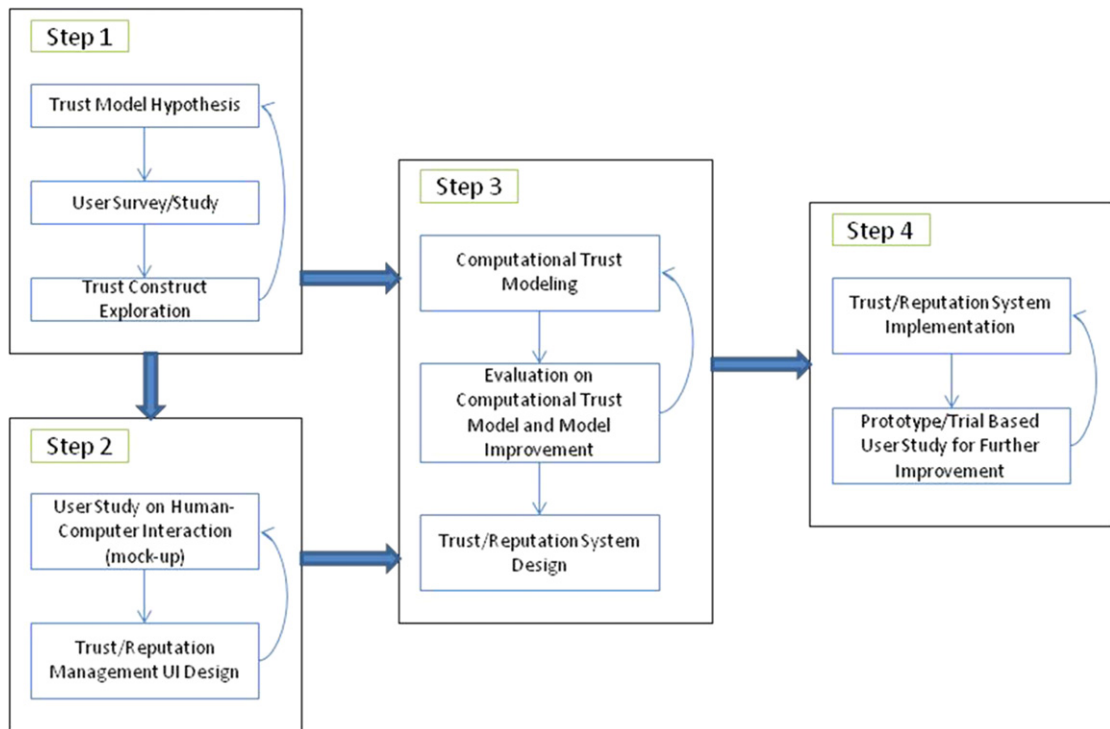


Fig. 1. A methodology of usable trust management development.

hard to trace malicious behaviors. It also influences the accuracy of reputation. Sun et al. proposed a number of schemes to overcome some of the above attacks, but they did not consider the additional challenges caused by privacy preservation [8]. PerChatRep aims to overcome the bad mouthing attack (i.e., unfair rating attack). Unfair ratings could be provided by dishonest, collaborative, and even profit-driven users. They don't rate a target entity truthfully or accurately. It is the most common and tough attack in the reputation system.

There are various methodologies applied for trust modeling and management. Some trust models are based on cryptographic technologies, e.g., Public Key Infrastructure (PKI) played as the foundation in a trust model [37]. A big number of trust models are developed targeting at some special trust properties, such as reputations, recommendations and risk [2]. In recent research, computational trust plays as a typical approach for trust modeling. In this approach, the characteristics of trust, principles or axioms are firstly presented; then they are modeled in a mathematical way; furthermore, the model is applied into trust evaluation or trust/reputation management for solving a specific issue. Examples of applying this method are the trust models based on Bayesian inference [38], weighted average, Dempster-Shafer theory [39,40], subjective logic [41], fuzzy logic [42], entropy-based models [9], etc. However, all above models are not driven by humans or users. Little research in this category evaluated user acceptance of proposed trust management systems. Thus it is hard to comment whether the systems are usable or not. Another approach of trust modeling aims to conceptualize trust based on user studies through a psychological or sociological approach (e.g., a measurement scale). This kind of research aims to prove the complicated relationships among trust and other multiple factors in different facets. The trust models generated based on this approach are generally linguistic or graphic. They do not quantify trust for machine processing purposes. Little work has been conducted to integrate psychological, sociological and computational theories together. In our opinion, the psychological and sociological study results can further play as a practical foundation for computational trust – modeling trust for a digital processing purpose. It would seem, therefore, that further investigations are needed in order to study usable trust management solutions.

Due to the subjective characteristic of trust, trust modeling and management should take users into concern. PerChatRep adopts a usable trust modeling and management methodology proposed by the lead author in [6] in order to make it easily accepted by the users towards practical deployment. This human-centric methodology is inter-disciplinary, which is different from the prior arts presented above.

### 3. Research method

The usable trust management methodology applied by PerChatRep contains four steps as shown in Fig. 1.

Step 1 – trust construct study. This step aims to study user concern on trust. Firstly, we conduct a user experiment with a measurement scale in order to find out the principal factors that influence the trust during pervasive social chatting. The

above procedure could be repeated in order to achieve a stable construct. A clear trust construct can be achieved based on a statistical analysis on collected experimental data. This result answers such a question as: what data or information should be considered in order to evaluate trust and/or reputation.

Step 2 – human–computer interaction (HCI) design for trust management. In Step 2, we conduct a relevant user study about human–computer interaction design for the trust management system. The user experiment could be mockup based. User feedback will be collected and analyzed in order to select a proper UI design solution, e.g., for reputation information visualization. Studying HCI for trust is important for us to understand the usefulness of trust and/or reputation visualization, the potential risk of visualization, the user preference of visualization control, and the user awareness of trust and reputation information.

Step 3 – computational trust modeling and system design. Step 3 aims to work out a computational user driven trust model and the design of trust/reputation management system based on the results achieved in Step 1 and Step 2. Notably, the trust construct achieved in Step 1 is a linguistic and graphic model, which cannot be directly applied into a digital computing system. In order to digitally manage trust, we should further work out a computational trust model on the basis of the trust construct achieved in Step 1. The computational trust model should concern the principle factors of the trust construct and their causal relationships. Importantly, it is essential to conduct laboratory simulations to evaluate its effectiveness and robustness. This is because it is hard to simulate various malicious behaviors or attacks in the user study and figure out the potential problems of the computational trust model. Through simulations, the computational trust model should be further improved and optimized. Based on the achieved trust model and HCI design, a complete trust management system with both back end and front end UI design can be achieved.

Step 4 – overall system evaluation. A prototype or a trial system is implemented. A user experiment can be performed in order to collect real system usage experiences and feedback. The system usability and easy acceptance can be evaluated in this step for further system improvement. The improved system can be re-evaluated for additional optimization.

It is important to note that some sub-steps listed above are iterative in order to achieve either a good model or a usable design. The purpose is to consider the preference and expectation of users as early as possible thus effectively saving the cost of the system development and enhancing user acceptance. In Step 1 and Step 2, we apply different designs of user studies and adopt different analysis methods to process user data for different purposes. For clarification, we separate them into two steps in the methodology illustrated in Fig. 1. The user-driven computational trust model plays as the core of the trust management system. It is different from traditional trust models. This lies in the fact that the model is achieved by formalizing an empirical trust construct in a mathematical way. It reflects user perspective and integrates the advantages of computational and social trust studies. Other user experimental studies can be further conducted in order to achieve trustworthy human–computer interaction that is required in the trust management system.

#### 4. Need assessment

Following the above method, we conducted a need assessment survey before designing and implementing PerChatRep in order to explore its potential usefulness, the principal factors that influence trust (i.e., trust construct) during pervasive social chatting and the user preference of reputation visualization. In order to save execution cost, we combined user studies in Step 1 and Step 2 together in the need assessment. We applied a 5-point Likert scale in the survey.

##### 4.1. Design

The survey contains three parts. The first part evaluates the potential significance of developing PerChatRep based on three chatting scenarios:

Scenario 1 (S1) – sharing the cost of ‘buy 3 pay 2’ goods in a shopping mall: Right now you are at a shopping center, and a product you want is on sale with a condition ‘buy 3 pay 2’. However, you only need one. You want to ask strangers nearby via your mobile phone whether he/she wants to share the discount with you.

Scenario 2 (S2) – sharing the price of a packet of 5 movie tickets in front of a movie theater: After shopping, you want to watch Avatar in a movie theater. The ticket price is 13.8 euro. However, if you buy a packet of 5 tickets, it will be 8.6 euro for each. You want to share the ticket packet with strangers nearby. You discuss whether he/she wants to share the discount with you via your mobile phone.

Scenario 3 (S3) – sharing a taxi ride after movie: After the movie, a lot of people are leaving the theater. You want to watch a figure skating competition quite far away. You would like to take a taxi and think about sharing a ride. You discuss with people nearby via your mobile phone whether he/she wants to share the ride with you.







The participants were asked to express their opinions on the usefulness of a reputation system in the above chatting scenarios.

The second part explores the factors that may influence the trust of a user in the pervasive social chatting. We examined 5 factors as defined in Table 1: 1) the crucial level of chatting topic (*cl*); 2) the depth of chatting (*dc*); 3) interest similarity (*is*); 4) on-chat voting (*cv*); and 5) voting-afterwards (*v*). We also designed a measurement scale (i.e., the items in Table 1) to measure these factors in the survey in order to justify which ones should be considered in PerChatRep. We asked the participants to mark their agreement on the items.

**Table 1**  
Factors influencing trust in pervasive social chatting.

Factors	Definition	Items
the crucial level of chatting topic ( <i>cl</i> )	The criticality of chatting content and purpose	1) I am generally cautious when chatting on a crucial topic. 2) It is easier for me to talk more in less crucial or important chat. 3) I am careful to disclose my personal information in a crucial chat.
depth of chatting ( <i>dc</i> )	The rounds of communications or interactions between two parties	4) I will chat more with others if I feel good to continue talking. 5) I need to chat with a person many rounds in order to make a decision. 6) It is important for me to ask a number of important questions during chatting in order to make a decision.
interest similarity ( <i>is</i> )	The common chatting communities shared between two parties	7) I feel safe to chat with a person if he/she chatted with me before. 8) I feel easy to chat with a person if I chatted with him before. 9) I feel nice if a person chat with me several things with common interests.
on-chat voting ( <i>cv</i> )	The vote on a specific message provided by a chatter during chatting	10) I would like to express my opinion on somebody's words during chatting. 11) I would like to know other people's opinion on a person during community chatting. 12) I trust a person more if I agree with his/her words more.
voting-afterwards ( <i>v</i> )	The vote on a party after chatting based on interaction experiences	13) It is helpful to know other people's opinion on a person after chatting. 14) I can evaluate a person more accurately after chatting than during chatting. 15) I can evaluate a person more accurately when I physically interact with him/her than without any physical interactions.

**Table 2**  
Reputation visualization methods.

UI1)	Based on the font size of the input text of a chatter: the bigger size of the font, the more reputable.
UI2)	Based on the number of stars, the more the higher reputation, e.g., ★★★★★☆
UI3)	Through a growing process of a cartoon character, the more mature the higher reputation, e.g.,  .
UI4)	Through a role in a community, a user can personally select or define the roles and their represented reputation levels, e.g., some characters from the Simpsons ( <a href="http://www.thesimpsons.com/">http://www.thesimpsons.com/</a> ).
	represents few reputable histories with trouble records;
	represents some reputable histories with trouble records;
	represents some reputable histories without trouble records;
	represents high reputation with some trouble records;
	represents high reputation without trouble records.

The third part attempts to study the user preference on reputation visualization. We proposed 4 visualization methods, as illustrated in Table 2: UI1 – reputation is indicated based on the font size of an input chatting message; UI2 – reputation is indicated by the number of stars; UI3 – reputation is indicated through a growing process of a cartoon character; UI4 – reputation is indicated through a role in a community, which can be customized by a user. We asked the participants to mark their preferences. Note that UI2 is a traditional reputation visualization method applied by Amazon and eBay.

#### 4.2. Participants and results

The survey was distributed through a mailing list. We conducted it in Finland and China and collected the survey response via email. A small gift was awarded to each participant. We got a total of 107 valid responses; among them 83 were university students, 68 (63.6%) male and 39 (36.4%) female, most participants were between 21–28 years old. All of them had Internet chatting experiences, 84.1% had mobile Internet chatting experiences, and 18.7% had experience on MANET-based pervasive social chatting. In our study, we used the samples mostly made up of university students. Note that it is very common for university students to use mobile phones in their routine life. Thus, the samples we adopted have certain universality. Future study would be useful to further prove the result with other representative samples.

The survey result and its implication are summarized as below:

1. The average rating scales regarding the potential usefulness of a reputation system in three pervasive social chatting scenarios were 3.74 ( $SD = 1.08$ ), 3.90 ( $SD = 0.98$ ) and 4.00 ( $SD = 1.00$ ), respectively. All of them are over 3.5. This implies

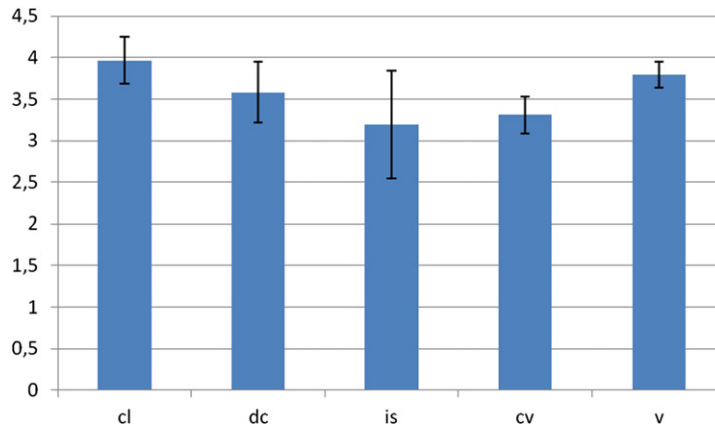


Fig. 2. Agreement scales on the factors influencing trust.

that a reputation system for pervasive social chatting is thought as useful in some instant social networking scenarios. Therefore, it is a significant contribution to design and develop such a system.

2. As shown in Fig. 2 (with standard deviation), the average agreement scales of the five factors (in Table 1) were ranged between 3.19 and 3.96. This implies that these five factors influence trust and should be considered in the design of PerChatRep reputation scheme. The survey provides us a good justification for considering these five factors in reputation generation for pervasive social chatting. Notably, the agreement scale of interest similarity (*is*) is lower than other factors, but it is still over 3.0. Thus, we still consider it in reputation generation for accurate evaluation.

3. Most participants preferred the traditional reputation visualization style UI2 (with an average value 4.07), but they were also interested in the new styles UI3 and UI4 proposed by us (UI3-3.32 and UI4-3.21). However, the font size-based reputation indication was not preferred (UI1-2.50). Some participants commented that UI4 design is very interesting and expect an implementation for optional selection. Some participants prefer other schemes of UI2 design (e.g., the number of crowns or diamonds). Thus, personalized reputation visualization is suggested in PerChatRep.

## 5. System design and implementation

### 5.1. Definitions

Before illustrating the reputation system structure, we firstly define three types of reputation provided by PerChatRep. Since PerChatRep applies a hybrid reputation system structure, the reputation can be generated by both the node and the trusted server. Thus, we have node local reputation, its general reputation and personalized reputation as defined below.

**Definition 1.** *Local reputation* of a node is the reputation value evaluated locally by another MANET node. The local reputation is independently evaluated by each node. Therefore, the local reputation values of the same node evaluated by different nodes may be different. We use  $R(i \rightarrow j)$  to denote node  $j$ 's local reputation evaluated by node  $i$ .

**Definition 2.** *General reputation* of a node is the reputation evaluated by the TS based on the collected social networking records. It is an attribute of a MANET node. It is evolved based on the social behaviors and the performance of the node voting-afterwards. The node can request the general reputation values of other nodes by requesting the TS. We use  $R(j)$  to denote node  $j$ 's general reputation issued by TS.

**Definition 3.** *Personalized reputation* is the reputation evaluated by the TS based on the social interaction experiences of one node on another node by considering the fifth factor – voting-afterwards. The node can request the personalized reputation values of other nodes from the TS. We use  $\overline{R(i \rightarrow j)}$  to denote the reputation of node  $j$  personalized for node  $i$  and issued by TS. It is the reputation value of node  $j$  from the view of node  $i$ , which is evaluated by TS.  $\overline{R(i \rightarrow j)}$  is different from  $R(i \rightarrow j)$  since  $R(i \rightarrow j)$  is the reputation of node  $j$  evaluated locally by node  $i$ .

### 5.2. System structure

Based on the survey results, we designed and developed PerChatRep. We attempt to utilize the advantages of both distributed and centralized reputation architecture. Fig. 3 illustrates PerChatRep structure. At each node device, a User Behavior Observer records node chatting behaviors. A MANET Social Networking UI provides a user interface for social networking. Communication Reporter and Voter report the social networking records and local reputation evaluation results to

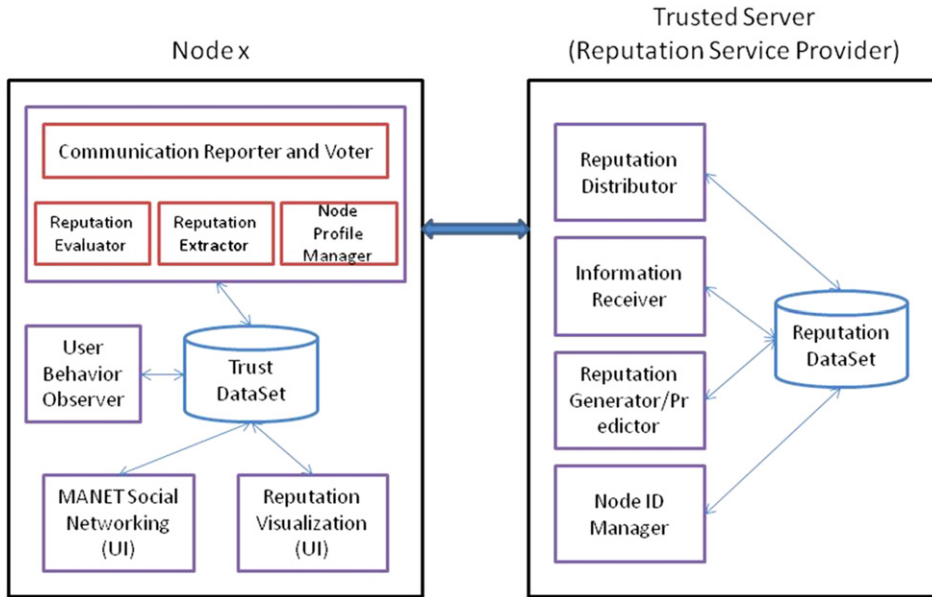


Fig. 3. PerChatRep system structure.

the Trusted Server (TS). Meanwhile, the user can also vote other nodes through it to TS. A Reputation Evaluator evaluates the local reputations of nodes and provides them to the user via a Reputation Visualization UI during chatting. A Reputation Extractor receives the reputation token that contains the node general and/or personalized reputation issued by TS. Trust DataSet securely stores all related data. A Node Profile Manager maintains the personal information of the user. It can communicate with TS to register the node into PerChatRep and update the pseudonym and reputation token of the user.

At the TS, a Reputation Generator/Predictor calculates the node general and personalized reputation values and identifies malicious nodes. A Reputation Distributor distributes the reputation tokens to each node periodically or by request. A Node ID Manager handles node registration and manages pseudonyms. An Information Receiver collects the records reported by the nodes and saves them into Reputation DataSet, which also saves the reputation tokens and the node real IDs and pseudonyms. Applying node pseudonyms in PerChatRep during chatting can enhance user privacy, but TS can still precisely provide general and personalized reputations based on historical social networking records according to real node identities. We try to overcome the inconsistent reputation problem in PerChatRep by introducing TS to unify the general reputation based on local experiences reported by nodes. This is achieved based on the system design in such a way that each node registers at TS with a unique ID, but the node can communicate with each other with frequently changed pseudonyms in PSN via MANET.

Introducing such a hybrid reputation system structure for PerChatRep has a number of merits. First, this design can support privacy preservation by frequently changing the pseudonyms of nodes and avoid inconsistent reputation problem. It can support accurate node reputation evaluation based on the registered unique node ID even though the node pseudonyms could be changed. Second, this design provides an economic approach to collect useful data from pervasive social networking that can further support other promising services, e.g., location-based content recommendation services. Thereby, the design can potentially support new business models. Finally and more importantly, this design is flexible to provide reputation information no matter TS is available or not. PerChatRep can generate reputation in either a distributed or centralized manner or both.

In PerChatRep, we assume that TS is trustworthy enough to preserve the private data of nodes. Notably, PerChatRep can also work in a distributed way based on the node pseudonyms when TS is not available.

### 5.3. Reputation generation

PerChatRep generates node reputation based on the following model. The TS generates/predicts node general/personalized reputation and issues a reputation token to the node. Based on the reputation token and pervasive social networking experience, a node generates the local reputations of other nodes during chatting. The local reputation values of nodes are further evolved based on their social networking behaviors and performance. Particularly, the node reports past chatting experiences or votes other users who chatted with him/her to TS. After collecting additional pervasive social networking records, TS updates the general/personalized reputation values for each node. The evaluation of node reputation is iterative at both the node and TS based on newly accumulated experiences and information.



To generate the local reputation  $R(i \rightarrow j)$ , we consider to aggregate three trust impact aspects together. The first part is the sum of previous local reputation (or personalized reputation if available) and the general reputation, which serves as the initial reputation of current chatting. The second part is the reputation generated on the basis of current chatting experience. Since there could be multiple on-chat votes during chatting, we integrate them together by averaging the product of on-chat votes and the depth of chatting at the vote that impacts the preciseness of the opinion of a node. Furthermore, this part is weighted by  $is(i, j)$  and  $cl(i)$ , which are other two on-chat factors influencing trust, as we explored in the need assessment. The third part is generated based on the on-chat votes on node  $j$  provided by other nodes than  $i$ , which is certified by the reputation of node  $k$   $R(i \rightarrow k)$  locally evaluated by node  $i$ . We apply formula (1) by considering the first four on-chat factors explored in the survey that are available at each node. This formula design follows the style of PeerTrust in [18], but with different factors considered for different purpose. Note that the simulation result in Section 6.1.2 shows the potential weakness of applying this formula with regard to collaborative bad mouthing attack.

$$R(i \rightarrow j) = f \left( \alpha (R'(i \rightarrow j) + R(j)) + \beta \sum_{l=1}^L \{cv(i \rightarrow j)_l * dc(i, j)_l\} * is(i, j) * cl(i) + \gamma \sum_{k \neq i} \sum_{l=1}^{L'} \{cv(k \rightarrow j)_l\} * R(i \rightarrow k) \right) \quad (1)$$

where  $R(i \rightarrow j)$  is the reputation of node  $j$  locally evaluated by node  $i$ .  $R'(i \rightarrow j)$  denotes the personalized reputation predicted and issued by TS, or the reputation previously evaluated by node  $i$  on  $j$ .  $R(j)$  is the general reputation of node  $j$  issued by TS.  $cv(i \rightarrow j)_l$  is the  $l$ th on-chat voting on the message of node  $j$  by node  $i$ .  $is(i, j)$  is the number of common communities shared by nodes  $i$  and  $j$ , which indicates their common interests.  $cl(i)$  is the crucial level of chatting topic.  $dc(i, j)_l$  is the depth of chatting between node  $i$  and node  $j$  at the time of the  $l$ th voting of node  $j$ . It is the minimum number of messages input by node  $i$  and node  $j$  at the time of the voting. For example, if node  $j$  has input 4 messages and node  $i$  has input 6 ones at the time of the  $l$ th voting of node  $j$  on  $i$  during their chatting,  $dc(i, j)_l = 4$ .  $f(x) = \frac{1}{1+e^{-x}}$  is the Sigmoid function used to normalize an arbitrary value into  $(0, 1)$ .  $L$  denotes the total number of on-chat votes by node  $i$  on node  $j$ .  $L'$  denotes the total number of on-chat votes by node  $k$  on  $j$ .  $\alpha, \beta, \gamma$  are parameters to indicate the weights of different contributions. Note that  $\alpha + \beta + \gamma = 1$ .

We further introduce an opinion deviation factor as in formula (2) and design formula (3) to generate the local reputation  $R(i \rightarrow j)$ .

$$od(i \leftrightarrow k, j) = 1 - 2 * \left| f \left\{ \sum_{l=1}^L cv(i \rightarrow j)_l * dc(i, j)_l - \sum_{l=1}^{L'} cv(k \rightarrow j)_l * dc(k, j)_l \right\} - \frac{1}{2} \right|, \quad (2)$$

$$R(i \rightarrow j) = f \left( \alpha (R'(i \rightarrow j) + R(j)) + \beta \sum_{l=1}^L \{cv(i \rightarrow j)_l * dc(i, j)_l\} * is(i, j) * cl(i) + \gamma \sum_{k \neq i} \sum_{l=1}^{L'} \{cv(k \rightarrow j)_l\} * od(i \leftrightarrow k, j) \right) \quad (3)$$

where  $od(i \leftrightarrow k, j)$  is the opinion deviation factor that indicates the difference of opinions between node  $k$  and node  $j$  on the chatting messages input by node  $i$  in the same chat. The opinion deviation factor indicates the opinion deviation of two nodes on a target node. Applying this factor makes it easy to figure out the nodes that hold different opinions from the reputation evaluating node. Thus applying it in reputation generation can avoid the negative influence of bad mouthing attack. This design is specialized for the scenario of pervasive social chatting based on our need assessment. Note that formulas (1) and (3) could be alternative, we select applying for formula (3) in PerChatRep if its user would like to use his/her own opinion to measure the trustworthiness of other people in the pervasive social chatting. As shown in our simulation in Section 6.1, formula (3) is effective to overcome collaborative bad mouthing attack.

Note that we can't assume most nodes are honest in pervasive social chatting. For generating  $R(i \rightarrow j)$ , we tailor the contribution of on-chat voting  $cv(k \rightarrow j)_l$  of other nodes based on two methods in order to ignore or reduce the influence of potential bad mouthing attack: one is applying the local reputation  $R(i \rightarrow k)$  to weight the on-chat voting of node  $k$  on  $j$ , shown in formula (1); the other is using the opinion deviation factor  $od(i \leftrightarrow k, j)$  to weight  $cv(k \rightarrow j)_l$ , shown in formula (3). In Section 6.1, we find that the second method is advanced in fighting against the collaborative bad mouthing attack through simulation-based evaluation.

Based on voting-afterwards and local reputation, we generate two types of node reputation at TS: personalized reputation and general reputation. We apply weighted aggregation using local reputation  $R(i \rightarrow j)$  as credibility to overcome unfair rating attack. Meanwhile, we also consider time influence on the voting-afterwards in order to overcome on-off and conflict behavior attacks [8].



Fig. 4. Create a chatting community.

Formula (4) is applied to generate the personalized reputation of node  $j$   $\overline{R(i \rightarrow j)}$  evaluated by node  $i$  by considering the fifth factor – voting-afterwards and time decaying. As specified in Table 1, the voting-afterwards is the vote on a node after chatting based on interaction experiences.

$$\overline{R(i \rightarrow j)} = \frac{1}{O} \sum_m R(i \rightarrow j)^{t_m} * V_i^{j(t_m)} e^{-\frac{|t-t_m|^2}{\tau}} \tag{4}$$

where  $O = \sum_m R(i \rightarrow j)^{t_m} * e^{-\frac{|t-t_m|^2}{\tau}}$ ;  $V_i^{j(t_m)}$  is the voting-afterwards of node  $i$  on node  $j$  at time  $t_m$ ;  $t$  is the calculation time of node reputation; parameter  $\tau$  (e.g.,  $\tau = 2$  in our simulations) is used to control time decaying.  $R(i \rightarrow j)^{t_m}$  is the local reputation of node  $j$  reported by node  $i$  at time  $t_m$ , with the voting-afterwards  $V_i^{j(t_m)}$  attached. If  $V_i^{j(t_m)}$  is not provided by the node, we set  $V_i^{j(t_m)} = 0.5$ . Note that  $V_i^{j(t_m)} \in [0, 1]$  and  $R(i \rightarrow j)^{t_m} \in [0, 1]$ .

To get the general reputation of node  $j$ , denoted as  $R(j)$ , we aggregate the evaluation  $\overline{R(i \rightarrow j)}$  of all nodes based on formula (5). The general reputation  $R(i)$  plays as the credibility of  $\overline{R(i \rightarrow j)}$  in the aggregation. Meanwhile, we also consider the influence of the number of reputation contributors on the general reputation generation, since the more contributors, the more convinced the generation result is.

$$R(j) = \frac{f(K)}{W} \sum_{i=1}^K R(i) * \overline{R(i \rightarrow j)} \quad (i \neq j) \tag{5}$$

where  $K$  is the total number of nodes who have direct experiences with node  $j$ .  $W = \sum_{i=1}^K R(i)$  is the total sum of the general reputation values of those nodes.  $f(K) = \{1 - \exp(\frac{-K^2}{2(\sigma+\epsilon)^2})\}$  is the Rayleigh cumulative distribution function to model the impact of  $K$  (i.e., the number of reputation contributors) on the node general reputation,  $\epsilon = -K/K'$ , is a factor to indicate sociability of node  $j$ . Parameter  $K'$  is the total number of registered users in the system.

#### 5.4. Implementation

PerChatRep is implemented based on a pervasive social networking platform [1]. This platform provides an energy-efficient and fully distributed social networking environment. We develop MANET nodes using Nokia N900 with Python and GTK binding. The MANET communications are based on wireless LAN. The TS is implemented with Apache and PHP in Linux platform (Ubuntu 9.04). The connection between the TS and nodes is based on wireless LAN or cellular networks. The implemented prototype system has three functional modules: pervasive social chatting, reputation management and privacy/identity management.

PerChatRep supports both node-to-node chatting and community chatting. Any user can create a community by indicating the community name and its importance (i.e., the crucial level of chatting topic) through the UI shown in Fig. 4. After creating a community, other people in vicinity can find the community in their device and join the community chatting. PerChatRep allows on-chat voting and reputation visualization during chatting. Fig. 5 shows a community chatting UI with personalized reputation visualization and on-chat voting with comments (e.g., “You DOWN Node 3: Too expensive” and “You UP Node 3: Good”). Particularly, PerChatRep user can select a preferred visualization scheme and activate or deactivate it. In Fig. 5, a reputation visualization scheme is shown with battery volume. PerChatRep also provides detailed information of personalized and general reputations by touching the ‘eyes’ icon and the user photos in Fig. 5.

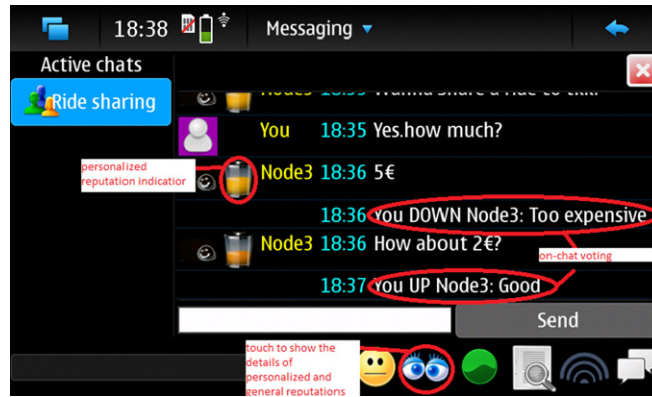


Fig. 5. Community chatting UI and on-chat voting.

Table 3

The chatting behavior data of node 4.

Voting No. $l$	$cv(i \rightarrow j)_l, i = 4, j = 5$	$dc(i, j)_l, i = 4, j = 5$
1	1 (positive up)	3
2	-1 (negative down)	5
3	1 (positive up)	7
4	1 (positive up)	9
5	-1 (negative down)	11
6	1 (positive up)	13

## 6. Evaluation results

Following our research method, we conducted simulations to evaluate the effectiveness and robustness of local reputation generation before implementing PerChatRep. We further conducted a prototype-based user study to show its usability and easy acceptance.

### 6.1. Algorithm evaluation based on simulations

We designed two simulations to evaluate the effectiveness and robustness of local reputation generation, using Matlab for both. For the robustness test, we focus on collaborative bad mouthing attack since we have showed that PerChatRep design can overcome other types of attack (such as on-off attack and conflict behavior attack) in our previous work [35].

#### 6.1.1. Effectiveness of PerChatRep

In Simulation 1, we set  $\alpha = 1/3$ ;  $\beta = 1/3$ ;  $\gamma = 1/3$ ;  $is(i, j) = 1$ ;  $cl(i) = 0.1$ ;  $R'(i \rightarrow j) = 0.5$ ;  $R(j) = 0.5$ ;  $L = 6$ ;  $K = 3$ ;  $L' = 6$ . There are at least five nodes in the system, i.e.,  $i = 4, j = 5, K = 3$ , which indicates nodes 1, 2, and 3 vote node 5's chatting messages. The chatting behavior data of node 4 are listed in Table 3.

We tested four cases when  $R(i \rightarrow k)$  is applied: (a) Case 1:  $R(i \rightarrow k) = 0.9$  (where  $k = 1, 2, 3$  and  $i = 4$ ); (b) Case 2:  $R(i \rightarrow k) = 0.5$  (where  $k = 1, 2, 3$  and  $i = 4$ ); (c) Case 3:  $R(i \rightarrow k) = 0.1$  (where  $k = 1, 2, 3$  and  $i = 4$ ); (d) Case 4:  $R(4 \rightarrow 1) = 0.9, R(4 \rightarrow 2) = 0.5, R(4 \rightarrow 3) = 0.1$ . In each of the above cases, we tested four sub-cases. In Sub-case 1, all other nodes vote the messages of node 5 as positive (i.e.,  $cv(k \rightarrow j)_l = 1$ ;  $k = 1, 2, 3$ ;  $j = 5$ ) at the same depth of chatting as the vote of node 4. In Sub-case 2, all other nodes vote the messages of node 5 as negative (i.e.,  $cv(k \rightarrow j)_l = -1$ ;  $k = 1, 2, 3$ ;  $j = 5$ ) at the same depth of chatting as the vote of node 4. In Sub-case 3, all other nodes vote the messages of node 5 the same as the node 4 at the same depth of chatting. In Sub-case 4, all other nodes vote the messages of node 5 differently from the node 4 at the same depth of chatting: node 1 always votes negatively (i.e.,  $cv(1 \rightarrow 5)_l = -1$ ); node 2 always votes positively (i.e.,  $cv(2 \rightarrow 5)_l = 1$ ); and the votes provided by node 3 are set as  $cv(3 \rightarrow 5)_l = \{-1, 1, 1, -1, 1, -1\}$ , where  $l = 1, 2, \dots, 6$ .

Fig. 6 shows Simulation 1 result by applying formula (1) when  $R(i \rightarrow k)$  is applied. Fig. 7 shows Simulation 1 result by applying formula (3) when  $od(i \leftrightarrow k, j)$  is applied. We observe that the reputation value in Sub-case 3 is the highest in Fig. 7 since all nodes hold the same opinions on the messages input by node 5. It is obvious that the deviation of opinions among nodes greatly impact the reputation value. The smaller the deviation, the higher reputation value is generated. Thus, applying formula (3) can fight against the collaborative bad mouthing attack as shown in Simulation 2 in Section 6.1.2. The reason that causes the difference of Fig. 6 and Fig. 7 lies in the fact that we consider the historical performance of other nodes reflected by  $R(i \rightarrow k)$  in formula (1); while in formula (3), we only pay attention to the node performance reflected by  $od(i \leftrightarrow k, j)$  in the current chatting.

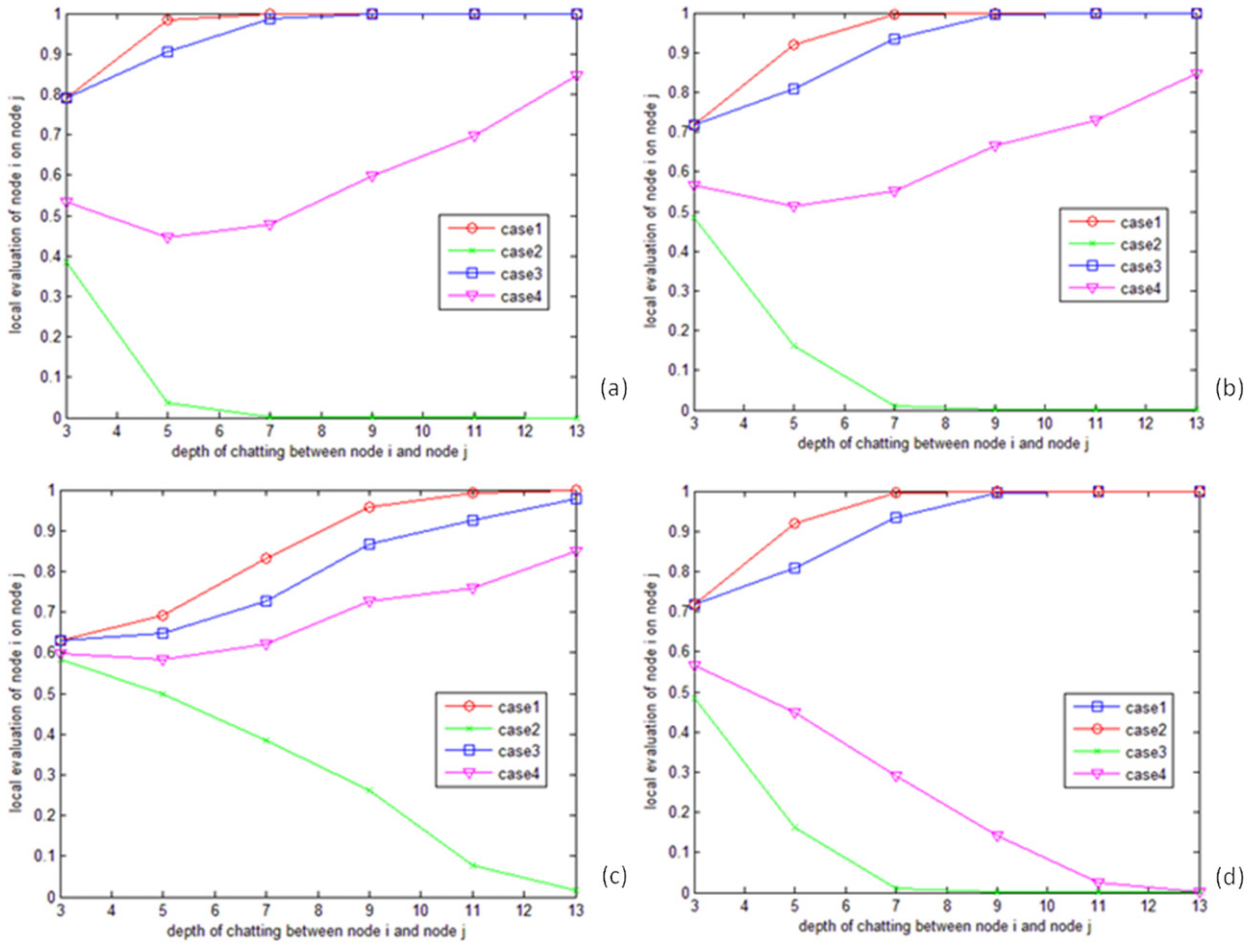


Fig. 6. Effectiveness of local reputation generation when  $R(i \rightarrow k)$  is applied. (a)  $R(i \rightarrow k) = 0.9$  (where  $k = 1, 2, 3$  and  $i = 4$ ); (b)  $R(i \rightarrow k) = 0.5$  (where  $k = 1, 2, 3$  and  $i = 4$ ); (c)  $R(i \rightarrow k) = 0.1$  (where  $k = 1, 2, 3$  and  $i = 4$ ); (d)  $R(4 \rightarrow 1) = 0.9, R(4 \rightarrow 2) = 0.5, R(4 \rightarrow 3) = 0.1$ .

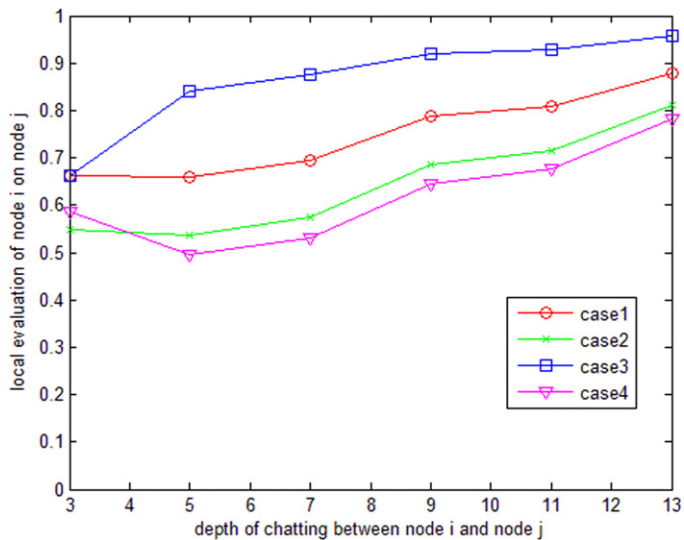
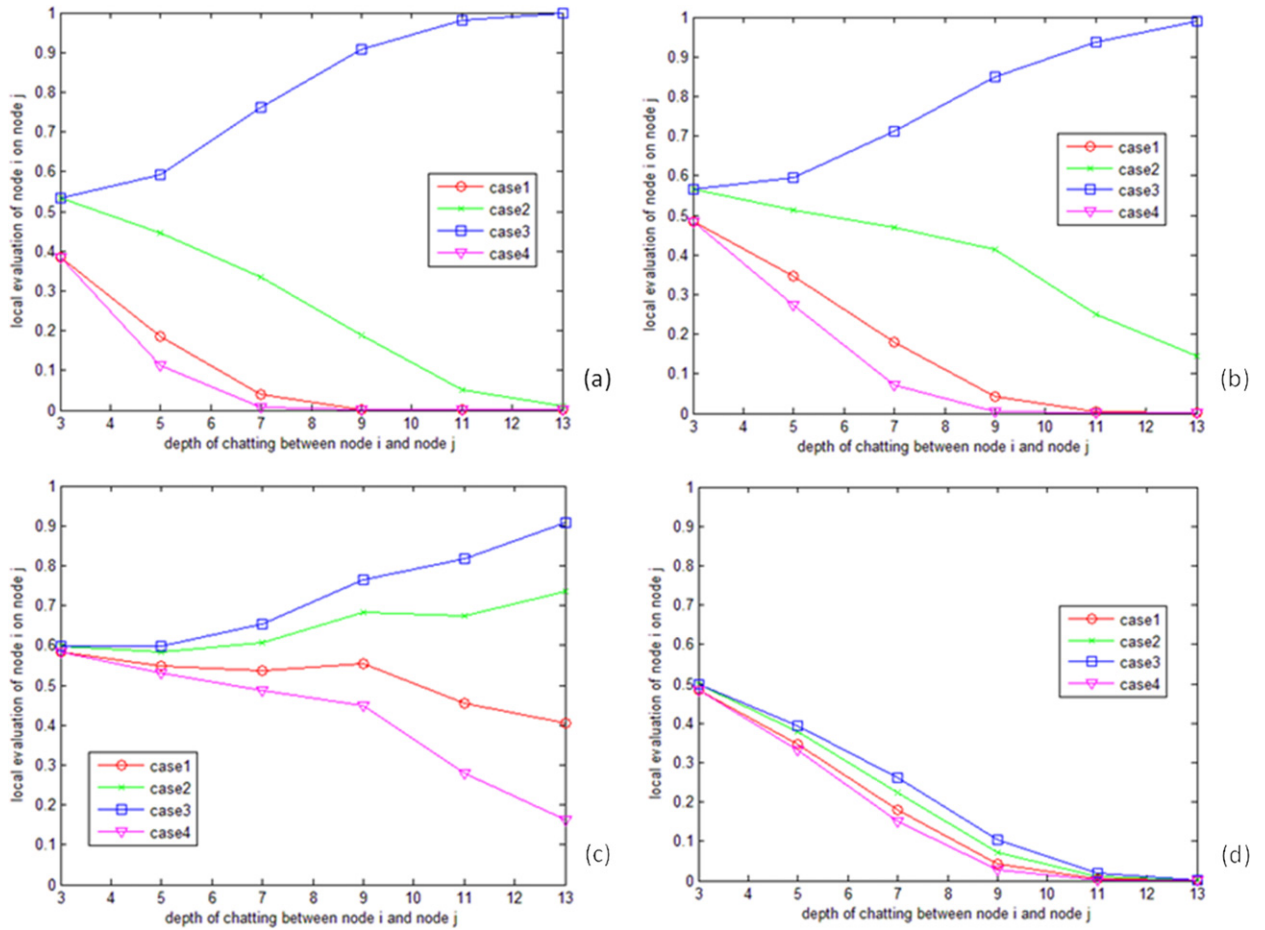


Fig. 7. Effectiveness of local reputation generation when  $od(i \leftrightarrow k, j)$  is applied.



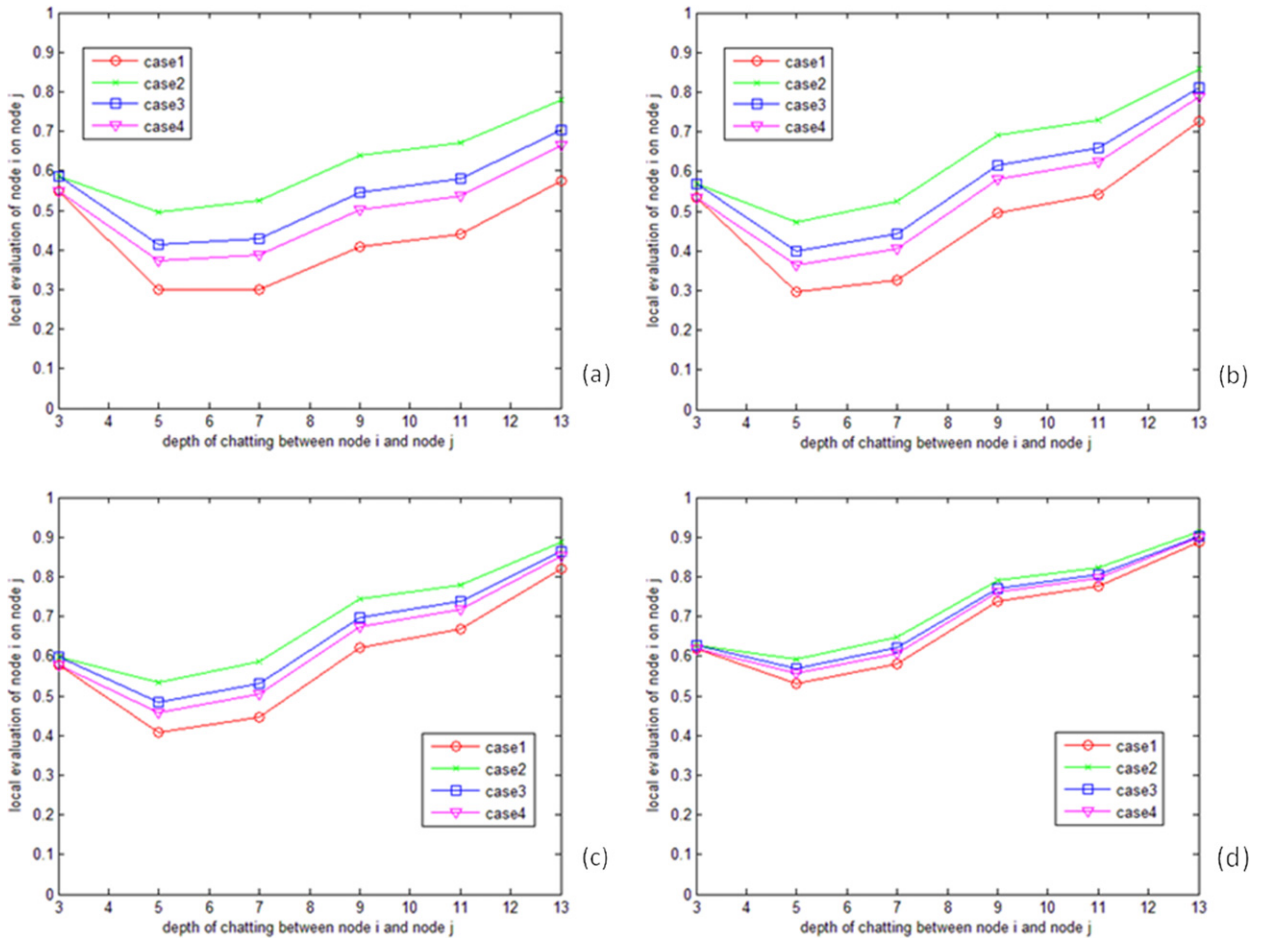
**Fig. 8.** Robustness of local reputation generation when  $R(i \rightarrow k)$  is applied. (a)  $R(i \rightarrow k) = 0.9$  (where  $k = 1, 2, 3$  and  $i = 4$ ); (b)  $R(i \rightarrow k) = 0.5$  (where  $k = 1, 2, 3$  and  $i = 4$ ); (c)  $R(i \rightarrow k) = 0.1$  (where  $k = 1, 2, 3$  and  $i = 4$ ); (d)  $R(4 \rightarrow 1) = 0.9, R(4 \rightarrow 2) = 0.5, R(4 \rightarrow 3) = 0.1$ .

### 6.1.2. Robustness of PerChatRep

We designed Simulation 2 to evaluate the robustness of local reputation generation under the collaborative bad mouthing attack. We kept the same simulation settings as in Simulation 1. We tested the same four cases as in Simulation 1 when  $R(i \rightarrow k)$  is applied, but the sub-cases show different types of collaborative bad mouthing attack: 1) all the nodes 1–3 vote oppositely from node 4 on the messages of node 5 at the same depth of chatting; 2) the nodes 1 and 2 vote oppositely from node 4, but the node 3 votes identically as node 4 on the messages of node 5 at the same depth of chatting; 3) nodes 1 and 2 vote oppositely from node 4, but the node 3 always votes positively on the messages of node 5 at the same depth of chatting; 4) nodes 1 and 2 vote oppositely from node 4, but the node 3 always votes negatively on the messages of node 5 at the same depth of chatting.

Fig. 8 shows Simulation 2 result by applying formula (1). We found that applying formula (1) may not overcome the collaborative bad mouthing attack when the local reputation value  $R(i \rightarrow k)$  of malicious node  $k$  is high. This could happen in PerChatRep when the malicious node  $k$  tries to achieve high local reputation  $R(i \rightarrow k)$  from node  $i$  by getting positive votes from node  $i$ . But node  $k$  could collaboratively decrease the reputation of another node  $j$ . In order to overcome this attack, we apply for formulas (2) and (3).

Fig. 9 shows the simulation result with four different settings of weight factors  $\alpha, \beta, \gamma$ . We observe that the local reputation  $R(i \rightarrow j)$  is mainly influenced by the votes of node  $i$ . The more common votes got from other nodes, the higher the local reputation is (e.g., in Sub-case 2). The opposite votes are filtered by the deviation factor, thus formula (3) is more effective against the collaborative bad mouthing attack than formula (1). We also found that the best way to reduce the influence of the bad mouthing attack is to reduce the value of parameter  $\gamma$ . We suggest setting  $\gamma = 0.1$  or  $\gamma = 0.2$  in practice. This strategy is also suitable for formula (1). Based on the simulation, we found the problem of formula (1), which is designed the same as many existing methods by using  $R(i \rightarrow k)$  as the credibility of on-chat voting [18]. In order to overcome the weakness of formula (1), we designed formulas (2) and (3) to fight against the collaborative bad mouthing attack. This result shows the advantage of our research method for figuring out potential risks of original design. In summary, we suggest applying formula (3) in practice.



**Fig. 9.** Robustness of local reputation generation when  $od(i \leftrightarrow k, j)$  is applied. (a)  $\alpha = 1/3; \beta = 1/3; \gamma = 1/3$ ; (b)  $\alpha = 0.2; \beta = 0.5; \gamma = 0.3$ ; (c)  $\alpha = 0.3; \beta = 0.5; \gamma = 0.2$ ; (d)  $\alpha = 0.4; \beta = 0.5; \gamma = 0.1$ .

6.2. System evaluation based on user experiment

We implemented PerChatRep and further tested its usability and user acceptance through real system usage.

6.2.1. Design

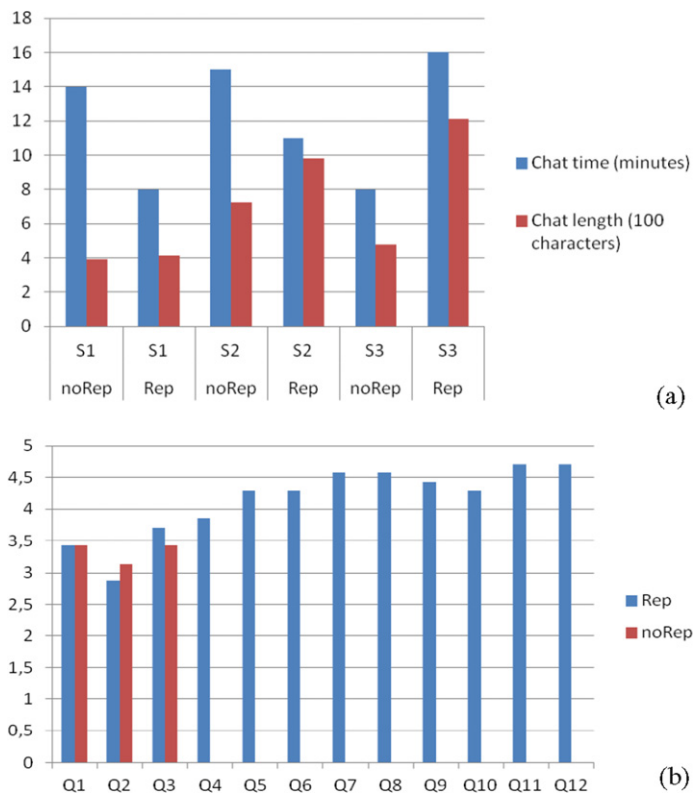
We performed a between-subject study to investigate the impacts of PerChatRep on mobile users. We selected 2 groups of participants from two student villages, each had 7 persons. All participants were university students aging between 23 and 28 years old. Among 7 participants in each group, 3 of them were female and 4 were male. They didn't know each other. All of them had Internet chatting experiences, but none of them had experience on pervasive social chatting. Group 1 used PerChatRep during chatting whilst Group 2 did not (i.e., turned off the function of reputation visualization).

We designed the experiment in a board game style in order to organize the study and make the results of two tests comparable. Before the experiment, each participant got a card that indicates his/her role and task in chat. We asked the participants to simulate three chatting scenarios as described in our survey. The participants tried to make a decision with regard to their chatting purpose. For each scenario, they chatted in a specific community. During the tests, the chatting information, such as chatting time, chatting contents, chatting length, on-chat voting, and voting-afterwards were automatically logged by PerChatRep for further analysis.

Additionally, we conducted an interview after the experiment to evaluate the perceived usefulness, perceived ease of use, interface, playfulness and user attitude in terms of PerChatRep. The participants were asked to express their agreement on the statements listed in Table 4. The participants in Group 2 only responded Q1–Q3. A 5-point Likert scale was applied. Our interview was designed based on the TAM model (Technology Acceptance Model) and its extension, which indicates that usefulness, ease of use and playfulness lead to user acceptance [14,15]. This theory also indicates that good interface leads to better perceived usefulness and ease of use; playfulness causes better acceptance (attitude). Finally, we randomly talked to some participants in order to get their additional comments. After the test, each participant was awarded a movie ticket.

**Table 4**  
Interview statements.

Purpose	Interview statements
Perceived ease of use	Q1: It is easy for me to start chatting with a person I don't know. Q2: It is easy for me to make a decision during chatting. Q3: It is easy for me to select a person I like from a number of candidates during chatting.
Perceived usefulness	Q4: PerChatRep can indicate user reputation appropriately during chatting. Q5: PerChatRep assists my decision on social networking during chatting. Q6: PerChatRep is a useful and helpful application.
Interface	Q7: Reputation visualization during chatting is useful. Q8: PerChatRep has a good design on reputation visualization. Q9: PerChatRep has a good design on reputation explanation. Q10: PerChatRep has a good design on user interface.
Playfulness	Q11: PerChatRep is an interesting application.
Attitude	Q12: I like using PerChatRep.



**Fig. 10.** (a) Chatting time and record length in three scenarios; (b) Average rates of interview statements.

### 6.2.2. Results and implications

Investigating the chatting time and length, we observe from Fig. 10(a) that displaying reputation information (i.e., 'Rep' case) could encourage participants to chat more and become more social (refer to chatting record length), and help them chat in a more efficient way (i.e., chatting time was shorter) than the situation without displaying reputation (i.e., 'noRep' case). We also note that participants became more serious and took longer time to make a decision in a more crucial chatting scenario (e.g., Scenario 3 – car riding), when the reputation value is visualized (refer to chatting time).

As shown in Fig. 10(b), PerChatRep has satisfactory evaluation scores with regard to perceived ease of use, perceived usefulness, interface, playfulness and user attitude. We got high average scores (>4.0) for Q5–Q12. In terms of perceived ease of use, we notify that visualizing reputation in pervasive social chatting made participants easier to select a person they like from a number of candidates than the case without reputation visualization. But there is no much difference in Q1–Q3 feedback between Rep and noRep cases. The result showed that PerChatRep is a very useful and interesting (playful) application that can aid user decision in pervasive social chatting. Its UI (especially reputation visualization) gained good

feedback from the participants. They liked using PerChatRep. Based on the TAM, we can conclude that PerChatRep was well accepted by the participants.

In addition, the random talks provided us interesting implications: (a) We found other potential use cases of PerChatRep such as dating and selling last-minute tickets at a concert hall; (b) The user interface of chatting log navigation need to be improved, e.g., displaying chatting messages in an anti-chronological order. This is because the participants felt difficult to browse long chatting messages; (c) Some participants regarded PerChatRep as a board game. They suggested extending PerChatRep to become a new platform for mobile gaming, where strangers nearby could play a game together.

## 7. Conclusions and future work

We designed and developed PerChatRep driven by the needs of mobile users following the methodology of usable trust management. The paper contributes in four-folds: (a) investigated how users consider and expect to cope with a reputation system for pervasive social chatting; (b) designed and developed a practical reputation system for pervasive social chatting, which is one of pioneer work in the literature; (c) verified its effectiveness and robustness through simulations and usefulness and acceptance through a prototype-based user experiment; (d) tested the impact of introducing and implementing PerChatRep into pervasive social chatting with reputation visualization.

Regarding the future work, we plan to do a trial to further improve PerChatRep and seek its business potential.

## Acknowledgments

This work is sponsored by the grant of Xidian University with the grant number K5051201032. The authors would like to thank Nokia Research Center. Part of this article work was conducted and sponsored by the Nokia Research Center, Helsinki.

## References

- [1] A. Ahtainen, et al., Awareness networking in wireless environments: means of exchanging information, *IEEE Veh. Technol. Mag.* 4 (3) (2009) 48–54.
- [2] Z. Yan (Ed.), *Trust Modeling and Management in Digital Environments: From Social Concept to System Development*, IGI Global, 2010.
- [3] MicroBlog, <http://synrg.ee.duke.edu/microblog.html>.
- [4] E. Sarigöl, O. Riva, P. Stuedi, G. Alonso, Enabling social networking in ad hoc networks of mobile phones, *Proc. VLDB Endow.* 9 (2) (2009) 1634–1637.
- [5] C.L. Corritore, B. Kracher, S. Wiedenbeck, On-line trust: concepts, evolving themes, a model, *Int. J. Human-Comput. Stud.* 58 (6) (2003) 737–758.
- [6] Z. Yan, V. Niemi, A methodology towards usable trust management, in: *Proceedings of ATC'09*, in: *Lect. Notes Comput. Sci.*, vol. 5586, 2009, pp. 179–193.
- [7] Familiar Stranger, <http://www.paulos.net/research/intel/familiarstranger/index.htm>, 2012.
- [8] Y. Sun, Z. Han, K.J.R. Liu, Defense of trust management vulnerabilities in distributed networks, *IEEE Commun. Mag.* 46 (2) (2008) 112–119.
- [9] Y. Sun, W. Yu, Z. Han, K.J.R. Liu, Information theoretic framework of trust modeling and evaluation for ad hoc networks, *IEEE J. Sel. Areas Commun.* 24 (2) (2006) 305–317.
- [10] G. Theodorakopoulos, J.S. Baras, On trust models and trust evaluation metrics for ad hoc networks, *IEEE J. Sel. Areas Commun.* 24 (2) (2006) 318–328.
- [11] Z. Yan, S. Holtmanns, Trust modeling and management: from social trust to digital trust, in: R. Subramanian (Ed.), *Computer Security, Privacy and Politics: Current Issues, Challenges and Solutions*, Idea Group Inc., 2008, pp. 290–323.
- [12] S. Trifunovic, F. Legendre, C. Anastasiades, Social trust in opportunistic networks, in: *IEEE INFOCOM Workshops*, 2010, pp. 1–6.
- [13] Z. Yan, C. Liu, V. Niemi, G. Yu, Effects of displaying trust information on mobile application usage, in: *ATC2010*, in: *Lect. Notes Comput. Sci.*, vol. 6406, 2010, pp. 107–121.
- [14] F.D. Davis, Perceived usefulness, perceived ease of use, and user acceptance of information technology, *Manage. Inf. Syst. Q.* 13 (3) (1989) 319–340.
- [15] V. Venkatesh, H. Bala, Technology acceptance model 3 and a research agenda on interventions, *Decis. Sci.* 39 (2) (2008) 273–315.
- [16] S. Song, K. Hwang, R. Zhou, Y.K. Kwok, Trusted P2P transactions with fuzzy reputation aggregation, *IEEE Internet Computing* 9 (6) (2005) 24–34.
- [17] P. Resnick, R. Zeckhauser, Trust among strangers in Internet transactions: empirical analysis of eBay's reputation system, in: M. Baye (Ed.), *Advances in Applied Microeconomics*, vol. 11: *The Economics of the Internet and E-Commerce*, Elsevier, 2002, pp. 127–157.
- [18] L. Xiong, L. Liu, PeerTrust: supporting reputation-based trust for peer-to-peer electronic communities, *IEEE Trans. Knowl. Data Eng.* 16 (7) (2004) 843–857.
- [19] J. Li, R. Li, J. Kato, Future trust management framework for mobile ad hoc networks, *IEEE Commun. Mag.* 46 (4) (2008) 108–115.
- [20] K. Walsh, E.G. Sirer, Fighting peer-to-peer SPAM and decoys with object reputation, in: *Proc. of P2PECON*, 2005, pp. 138–143.
- [21] A. Jøsang, R. Ismail, C. Boyd, A survey of trust and reputation systems for online service provision, *Decis. Support Syst.* 43 (2) (2007) 618–644.
- [22] M. Raya, P. Papadimitratos, V.D. Gligory, J.-P. Hubaux, On data-centric trust establishment in ephemeral ad hoc networks, in: *Proc. of IEEE INFOCOM*, 2008, pp. 1238–1246.
- [23] S. Buchegger, J.L. Boudec, Performance analysis of the CONFIDANT protocol, in: *Proc. of the ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, 2002, pp. 226–236.
- [24] P. Michiardi, R. Molva, CORE: a Collaborative REputation mechanism to enforce node cooperation in mobile ad hoc networks, in: *Advanced Communications and Multimedia Security*, in: *Lect. Notes Comput. Sci.*, vol. 2828, 2002, pp. 107–121.
- [25] L. Xiong, L. Liu, A reputation-based trust model for peer-to-peer e-commerce communities, in: *Proc. of the IEEE Conference on E-Commerce*, 2003, pp. 228–229.
- [26] S. Buchegger, J.Y.L. Boudec, The effect of rumor spreading in reputation systems for mobile ad-hoc networks, in: *Proc. of WiOpt Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, 2003.
- [27] Junction. Harvard MobiSocial Group, <http://openjunction.org/>.
- [28] P. Resnick, K. Kuwabara, R. Zeckhauser, E. Friedman, Reputation systems, *Commun. ACM* 43 (12) (2000) 45–48.
- [29] K. Kujimura, T. Nishihara, Reputation rating system based on past behavior of evaluators, in: *Proc. of the 4th ACM Conference on Electronic Commerce*, 2003, pp. 246–247.
- [30] S. Bansal, M. Baker, Observation-based cooperation enforcement in ad hoc networks, Technical report, Stanford University, NI/0307012, 2003.
- [31] J. Hu, M. Burmester, LARS: a locally aware reputation system for mobile ad hoc networks, in: *Proc. of the 44th ACM Annual Southeast Regional Conference*, 2006, pp. 119–123.



- [32] [Y. Yang, Y. Sun, S. Kay, Q. Yang, Defending online reputation systems against collaborative unfair raters through signal modeling and trust, in: SAC'09, 2009, pp. 1308–1315.](#)
- [33] [J.R. Douceur, The Sybil attack, in: IPTPS, in: Lect. Notes Comput. Sci., vol. 2429, 2002, pp. 251–260.](#)
- [34] [Z. Liu, S.S. Yau, D. Peng, Y. Yin, A flexible trust model for distributed service infrastructures, in: Proc. of 11th IEEE Symposium on Object Oriented Real-Time Distributed Computing, 2008, pp. 108–115.](#)
- [35] [Z. Yan, P. Zhang, R.H. Deng, TruBeRepec: a trust-behavior-based reputation and recommender system for mobile applications, Personal and Ubiquitous Computing 16 \(5\) \(2012\) 485–506, <http://dx.doi.org/10.1007/s00779-011-0420-2>.](#)
- [36] [Z. Yan, Y. Chen, AdContRep: a privacy enhanced reputation system for MANET content services, in: UIC2010, in: Lect. Notes Comput. Sci., vol. 6407, 2010, pp. 414–429.](#)
- [37] [R. Perlman, An overview of PKI trust models, IEEE Netw. 13 \(6\) \(1999\) 38–43.](#)
- [38] [J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, Morgan Kaufmann, 1988.](#)
- [39] [A.P. Dempster, A generalization of Bayesian inference, J. R. Stat. Soc. Ser. B 30 \(1968\) 205–247.](#)
- [40] [G. Shafer, J. Pearl \(Eds.\), Readings in Uncertain Reasoning, Morgan Kaufmann, 1990.](#)
- [41] [A. Jøsang, A logic for uncertain probabilities, Internat. J. Uncertain. Fuzziness Knowledge-Based Systems 9 \(3\) \(2001\) 279–311.](#)
- [42] [A. Tajeddine, A. Kayssi, A. Chehab, H. Artail, Fuzzy reputation-based trust model, Applied Soft Computing 11 \(1\) \(2011\) 345–355.](#)
- [43] [EZSetup, <http://research.microsoft.com/en-us/groups/wn/mssn.aspx>.](#)
- [44] [Nokia-instant-community, <http://conversations.nokia.com/2010/05/25/nokia-instant-community-gets-you-social/>.](#)