Electronic Commerce Research manuscript No.
(will be inserted by the editor)

CellTrust: a Reputation Model for C2C Commerce*

Gianluca Lax, Giuseppe M.L. Sarné

DIMET Dept.,
University of Reggio Calabria
Feo di Vito, 89122 Reggio Cal., Italy
e-mail: lax@unirc.it, sarne@unirc.it

Received: date / Revised version: date

Abstract The improvement of wireless technologies and the increasing spread of mobile phones open new possibilities to perform mobile Customer-to-Customer commercial activities. In this new scenario, where users cannot rely on stable connections, it assumes a great relevance how to trust the counterpart in a transaction and how to avoid that a disconnection, possible in wireless connections, can encourage users to cheat. To tackle these issues we propose a feedback-based reputation mechanism able to detect malicious users better than other state-of-the-art techniques, as shown by the large number of experiments run to measure the accuracy of the compared methods in the most common situations.

Key words Reputation – C2C commerce – Mobile Commerce

1 Introduction

The recent improvements of both wireless technologies and mobile devices will allow the mobile-commerce to become a dominant form of trading in the next future. The last wireless technologies like the 802.11 family [19] and Bluetooth [20] have reached good results in terms of operating range, hardware costs and power consumption so that users can communicate in a wide range with a high data rate. Consequently, they will be able to realize wireless personal networks in a peer-to-peer fashion without exploiting telephonic or networking providers. This opens new trading scenarios where users provided with their mobile phones can autonomously perform

* An abridged version of this paper appeared in Proceedings of the "ICE-B International Conference on E-Business 2006 - (ICE-B 2006)" [29]

Customer-to-Customer (C2C) activities [4]. This new possibility of trading raises a number of concerns regarding the security of transactions that in this scenario becomes very critical [33,51,56]. Think of a user supplied with a mobile device who buys a digital song, downloading it to his device from another user while they are in the subway. Threats just derive from the use of a decentralized architecture with temporary plug-in connections in which trading occurs. For example, during the trading (1) a user could be unable to contact a central server or other users in order to obtain information about the counterpart's trustworthiness or (2) the connection can abort (also fraudulently) before the transaction is concluded. These two last issues surely represent relevant problems characterizing such a *dangerous* scenario from the point of view of fraud occurrences, that are encouraged by anonymity of users involved in trading [52].

The usual approach to face these problems consists in splitting the trading in two phases, the first is carried out autonomously by the users, while the second is realized off-line via the Internet by a central entity that really commits the transaction. As a consequence, frauds can be detected only at the end of the transaction process when users have already spent a lot of precious resources, for example the battery capacity, that in a mobile context are limited.

Another possible solution consists in preventatively providing users with suitable information about the possibility to carry out or not a trading with another user. In particular, the knowledge about the past history of a user can represent his trustworthiness by means of the concept of *reputation* [30,31,47]. Obviously, such a reputation has to be spread among users to be meaningful. Typical approaches adopted in distributed environments assume that reputation values are propagated by users during their interactions [23,34,35]. Such approaches cannot be exploited in our context because they assume that (1) networks are stable enough both in composition and in time living and because (2) the acquired knowledge of information and reputation might be broadly partial and consequently ineffective in wide and dynamic communities.

In this paper we address the issue of security in this scenario. We analyze several actions that malicious users could do in order to gain reputation or to cheat or to disturb trading of honest users. In consideration of this analysis, we design a new reputation model, named *CellTrust*, to contrast such malicious activities and we describe how to compute and spread users' reputation in our scenario characterized by temporary connections.

The main contributions of our paper are:

- A new reputation mechanism is proposed in order to compute the reputation of users. Unlike other proposals, it takes into account some new parameters to avoid that malicious users cheat or disturb trading of other users. We have experimentally evaluated the proposal in order to verify its performance with respect to other reputation systems. The experiments have confirmed its capability to support C2C trading activities by detecting malicious users better than other relevant techniques.

– The proposed solution tackles and solves numerous problems that come out from this scenario, such as how to identify the counterpart and how to know his reputation without contacting any server before the trading. Besides, our proposal takes into account and satisfies the requirements defined in a recent paper [33] to guarantee the security of a mobile transaction.

The plan of the paper is the following. In the next section we introduce the scenario we will consider along the paper. The discussion about several malicious activities, the parameters characterizing our reputation model as well as the reputation metrics are provided in Section 3. In Section 4 we discuss the issues of cost and scalability of the proposal. The comparison with related work is presented in Section 5. The results of the experiments done to compare our reputation model with two other approaches are reported in Section 6. Finally, in Section 7 we draw our conclusions.

2 Reference Scenario

In this section we describe the C2C scenario that could exploit our reputation proposal in order to detect cheaters and to reduce the possibility of being victim of frauds. The actors of this scenario are users who exploit their cell phone and a wireless connection to exchange digital resources and a *Reputation Manager* which manages users' reputation.

The overall process of trading consists of four steps that are described in detail below and schematized in Figure 1.

1. **Affiliation.** All users belonging to our system must be registered. This is done in the affiliation steps, where each user sends an affiliation request to the Reputation Manager via an Internet connection (for example UMTS or GPRS). The Reputation Manager is responsible for the user identification that is done by exploiting the SIM or the mobile phone number of the user's cell
   phone [33]. As a consequence, each identity change necessarily requires another SIM. Observe that in a real implementation, the Reputation Manager job might be carried out by the mobile telecommunication provider of the user, that has already identified its costumers. In this case, the affiliation to the system can be viewed as the activation of a new service (supplied by the provider) allowing the user to perform mobile commerce activities.
   After user identification, the Reputation Manager provides the user with a *trading tool*, consisting of the following data:
   – A pair of asymmetric cryptographic keys, exploited by the user to sign messages he sends and by the other users to verify the authenticity of the received messages.
   – A *(reputation) credential*, exploited by the user during transactions to certify his reputation. This credential stores the user's identifier,

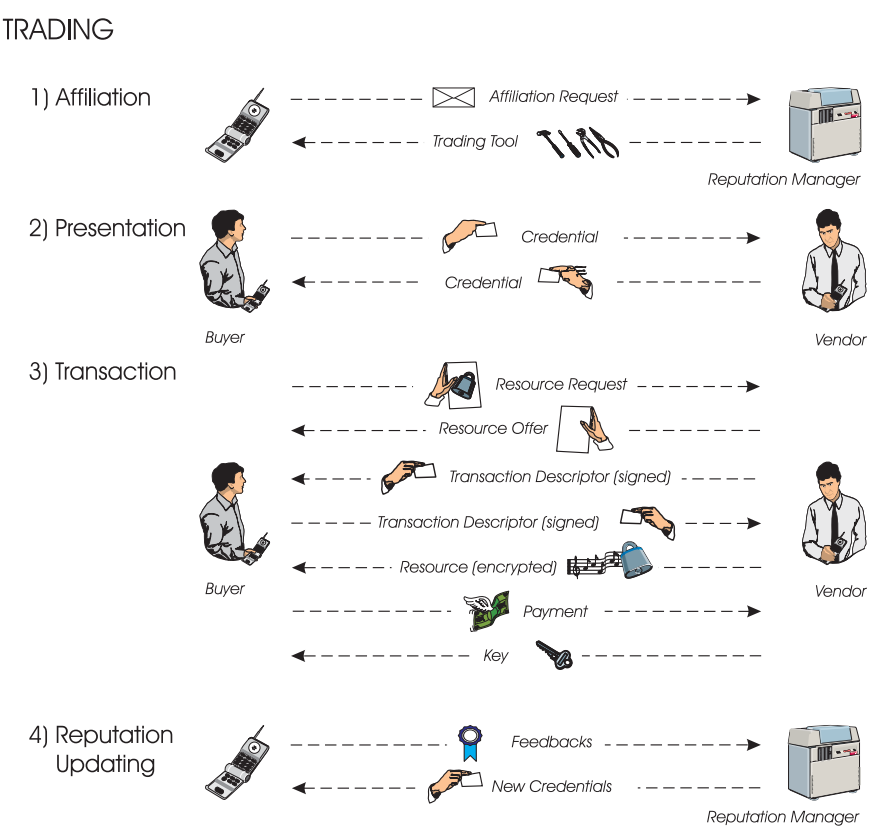


Fig. 1 C2C trading in our reference scenario.

his reputation rating ranging from 0 to 1, and the expiration date of the credential. Moreover, the credential is signed by the Reputation Manager in order to guarantee the integrity and the authenticity of the information reported.

– The digital certificate of the Reputation Manager. The user can validate the signature of the credential of other users by exploiting the public key it contains.

Concerning the reputation credential, an important parameter is the expiration date. Indeed, during the validity of the credential, the reputation of the user could change and, consequently, be different from the one reported in his credential. This problem is common to all the systems based on certificates. Different approaches have been proposed to tackle this issue. The most common one exploits *Certificate Revocation List* (CRL) [17], a list of certifications no longer valid, which is managed by the certification authority and can be publicly consulted. An alternative to CRL is based on *Online Certificate Status Protocol* (OCSP) [37], which can provide more timely information regarding the

revocation status of a certificate without burdening the network. However, such approaches would require the user to contact the Reputation Manager before each transaction. Other solutions are based on threshold cryptographic security schemes [28], hierarchical and distributed trust models [3,10], and allow users to distribute the list of good or bad users without accessing the Reputation Manager.
Among the numerous solutions that could be exploited to verify the validity of a credential, the approach we follow in our proposal is to issue credentials with a limited validity in order to decrease the likelihood that the reputation value stored in the credential be very different from the actual one. Even though cannot set to zero the probability that a fraud occurs, this simple solution has the advantage of allowing users to complete the transaction without the need to contact the Reputation Manager or other users.

2. **Presentation.** After the affiliation, the C2C trading between users is done by a wireless connection, like Bluetooth for example, as follows.
When two users are quite near to communicate by a wireless connection, the following operations are performed.
Each user sends his credential to the other one. Then, he verifies that the other is a *trustworthy* user by checking that the value of the reputation, extracted from the received credential, is higher than a personal *hazard threshold* [14,42,55], a value representing the transaction failure probability that a user is willing to accept. If this check fails, then the communication ends and the next operations are not performed.
At the same time, also the other user does the same check. If both users are considered (in the counterpart's opinion) trustworthy, the next step starts.
3. **Transaction.** In this step, without lost of generality, we consider that the two users introduced above are a buyer and a vendor. The former sends the latter a description of the resource he wants to buy and the vendor returns the list of resources matching the request as well as their price. If there is not a matching between offer and demand, then the process ends here. In case the purchaser wants to buy at least one resource, the transaction can continue.
Then, for each of such resources, the seller produces a *transaction descriptor*, a digital document reporting the buyer's ID, the seller's ID, a description of the resource and its price. This transaction descriptor is signed and sent to the counterpart by both users in such a way that (1) by verifying its signature each user can make sure both the counterpart identities and (2) the terms of the transaction (object to sell and price to pay) are fixed and cannot be repudiated – this descriptor could be exploited to solve possible complaints.
At this point the vendor sends the buyer the resource encrypted by a randomly chosen symmetric key. Observe that the resource is sent in a ciphered way in order to decrease the likelihood that an interruption of the transaction could allow the buyer to receive the resource without

payment. Indeed, the probability of disconnection raises proportionally to the connection duration. Since the size of the file to be sent is much larger than that of the other messages exchanged, from the point of view of a possible disconnection, the resource transmission is the most crucial moment of the transaction. If a disconnection occurs before the conclusion of the file transfer, then the transaction aborts, and all the operations done have no meaning. Otherwise, that is when the encrypted resource is sent successfully, the buyer sends the payment code [38,39] and the vendor has to send the symmetric key used to encrypt the file. The transaction between the two users is thus concluded.

4. **Reputation Updating.** Periodically, each user contacts via an Internet connection (for example GPRS or UMTS) the Reputation Manager to deliver all the transaction descriptors signed from the counterpart. On the basis of the transactions performed, the Reputation Manager updates the users' reputations according to reputation metrics defined in Section 3.2 and the user receives a new credential with an updated value of reputation.
   In this step, each user also provides a feedback for each transaction, a score of the reliability of the counterpart. The Reputation Manager exploits such feedbacks to update users' reputations according to the reputation model presented in the next section.

3 The Reputation Model

In this section, we give the motivations for our reputation system and then we define its metrics. In the following, according to [1], we consider reputation as "*an expectation about the user's behavior based on information about or observations of his past behavior*".

3.1 Motivations

When economical interests are relevant, like in electronic commerce, actors should be trustworthy. To this end, different technologies can be exploited, each one with different costs [27,40]. In every case, reputation information, based on a direct and/or indirect knowledge, is very important. Dealing with indirect information, the credibility of information sources assumes a great relevance [15].

In this paper we propose a feedback-based reputation mechanism, named *CellTrust*, expressly designed for the scenario we are considering. Indeed it takes into account some parameters that, in our opinion, are relevant in this context but are often ignored by other approaches. In particular, in our scenario a good reputation mechanism should satisfy the following properties:

1. Taking into account the trade history of each user;

2. Differentiating dishonest from honest feedbacks, to avoid malicious reputation manipulations;
3. Detecting collusive behaviors trying to increase malicious user's reputation or to discredit good users;
4. Recognizing different transaction contexts, to avoid reputation gain in small-value transactions for cheating in high ones;
5. Penalizing adversary's activity aimed to disturb trading, even when such an activity does not give the adversary any evident advantage.

The first four features are only partially considered in other reputation systems, as we will show in Section 5. The latter one is an aspect that, to the best of our knowledge, is new in the field of reputation systems and assumes a great importance in our scenario as we will describe below.

Our reputation system tries to satisfy the above five requirements by exploiting the following five parameters.

1. The first parameter takes into account the user's history and is based on the number of transactions done and the value of feedbacks obtained by the user. However its relevance should be limited in order to avoid that a user who gained a high reputation by a high number of good transactions, might cheat occasionally relying on the fact that a small number (with respect to the whole number of transactions effected) of negative feedbacks does not corrupt his reputation.
   In order to contrast these malicious attempts, we follow a simple but effective approach. The system is designed in such a way that negative feedbacks decrease the reputation value much faster than positive feedbacks increase it, so that a user loses his good reputation after some negative feedbacks. Obviously, on the other hand, this choice could be too penalizing for a user who is victim of false negative feedbacks. The use of the next parameter helps us to avoid this occurrence.
2. The parameter FC (*Feedback Credibility*) denotes the credibility of the feedback source that, as remarked in the previous item, is used to avoid that false feedbacks (possibly derived from collusive attempts) can alter the user's reputation. To this end the reputation of the user providing the feedback has to be considered to tune the weight given to the feedback. At the same time, we have to contrast that a user, exploiting his high reputation, could spread negative feedbacks to discredit other users. Regarding this last issue, observe that in theory we can assume that a bad transaction and, consequently, a negative feedback should occur very rarely in an environment where a good reputation system works. Also the well-known scenario of eBay respects this observation since, as shown in [46], solely about 0.6 percent of feedbacks provided by eBay users are negative.
   As a consequence, in CellTrust the weight of a negative feedback given by the user U is proportional to his reputation and inversely proportional to the number of negative feedbacks provided by U in the past.

In such a way, negative feedbacks provided by "habitual" complainers are neglected.
Note that the introduction of this parameter tackles also a typical problem of feedback-based approaches, and in particular of eBay, called *feedback blackmail* [5,26,32,45,47,48], in which a user threatens negative feedbacks to gain an unfair concession. Because of such a blackmail, many users automatically leave a positive feedback, endangering the whole reputation mechanism.

3. The parameter C (*Collusion*) is adopted to avoid that a user could affect too much another member's reputation or that two (or more) users could increase their reputation mutually by selling and buying resources. For this purpose different solutions have been proposed in the literature. The most popular one is used in eBay [21] where a member can increase or decrease another member's score only one time, no matter how many transactions they share. The solution adopted by eBay appears too strict in our scenario. Here transactions occur between users who are physically close (consider that due to the range of the wireless technology, users involved in a transaction usually live in nearby houses, work in the same company, take the same bus, and so on). In particular, transactions show a strong *locality* creating thus more *clusters* of traders and we can expect that users belonging to the same cluster and sharing similar interests may do more transactions together.
   Observe that this locality is a specific characteristic of our scenario and cannot be easily found in other environments (for instance, the likelihood that users of different cities can trade by eBay is not null, whereas such a probability is null in case the trading must be performed by a Bluetooth connection). For this reason, traditional approaches aimed to contrast collusive behaviors can fail in our case.
   Thus we introduced the parameter C in such a way that the relevance of the feedback provided by a user U_i in updating the reputation of a user U_j is inversely proportional to the fraction of transactions performed by U_j having U_i as counterpart. This parameter plays a role similar to the support defined for data mining association rules: it gives a measure (in terms of number of feedbacks) of how much the transactions performed with U_i characterizes the reputation of U_j.
4. The parameter TV (*Transaction Value*) is exploited to prevent a user from adapting his behavior to the value of the transaction to gain credibility by well behaving in low-cost transactions and, then, cheating in high-priced transactions. In our reputation model, the weight of feedback to reputation computing is proportional to the value of the transaction, so that low-cost transactions can increase reputation slightly whereas high-value ones affect reputation heavily.
5. The last parameter N (*Noise*) is used to penalize user activity aimed to disturb trading. Here we refer to the possibility that a malicious user could start a trading and then aborts it. This activity does not give malicious users any advantage but they have to be penalized because

it produces a damage to honest users who unfruitfully spend resources, like the battery of the device, very precious in mobile environments. This parameter is computed by considering the number of transactions started by the user with respect to the whole number of transactions successfully done.

3.2 The Reputation Metrics

All the above considerations can be expressed in a mathematical form. We assume that the reputation score R of a user is a floating number ranging from 0 to 1, such that the higher the value of R, the better the user's reputation is. The initial reputation score of a new user is fixed to 0.5. This choice takes into account two opposite requirements: the former is not penalizing new users [43], the latter is penalizing the user having a bad reputation who wants to reenter the system with a new reputation [59]. In fact, since reputation is an index of the probability that a user will behave well in the next transaction, in the case of a new user, whose history is not known, we can set such a probability to 0.5.

In order to show how the reputation of each user is computed and updated, we define a transaction T performed by the user U_i with the user U_j (called also counterpart in the following) as a tuple $\langle ID_j, val, f_j \rangle$ where: ID_j is the identifier of U_j, val is the monetary value of the transaction describing the transaction context (i.e., its relevance), and f_j is the feedback provided by U_i that represents his appreciation about the transaction (we assume it is 0 for unsatisfying transactions, 1 for full satisfaction). As a consequence of this transaction the reputation R_j of U_j is updated as follows:

$$R_j = (1 - \alpha) \cdot \widetilde{R_j} + \alpha \cdot f_j$$

In the equation, the new value of the reputation R_j of U_j is computed weighting in a complementary way, by means of α, a value ranging from 0 to 1, two contributions, (1) the previous reputation value of U_j (denoted by $\widetilde{R_j}$) and (2) the feedback provided by U_i. The variable α allows us to tune the two components.

Concerning the value of α, it is computed by suitably combining the parameters FC, TV, and C previously described as follows:

$$\alpha = \big((1 - \beta) \cdot FC + \beta \cdot TV \big) \cdot C \quad \text{with} \quad 0 \leq \beta \leq 1$$

The parameter β weighs the relevance of TV and FC to reputation updating, whereas the parameter C reduces α in case of collusive activity. Observe that in many cases FC and TV have the same relevance so that we can set $\beta = 0.5$. Now, let us define the three parameters used above.

The first one is Feedback Credibility $FC = \frac{R_i}{R_j + R_i} \cdot \left(1 - \frac{f_i^{\ominus}}{f_i^{\ominus} + f_i^{\oplus}}\right)$. In words, it is high whenever (1) the counterpart has a high reputation (denoted by R_i) with respect to the reputation of U_j (denoted by R_j) and (2)

the number of negative feedbacks provided by U_i ($f_i^{\ominus}$) is negligible with respect to the total number of feedbacks ($f_i^{\ominus} + f_i^{\oplus}$) provided by U_i.

For the parameter Transaction Value we have $TV = \frac{T_v}{v_m}$, that is close to 1 if the value of the transaction T_v is considerable compared with the maximum value v_m of all transactions. Note that this parameter is specially designed for our scenario and can be well defined in this context where resource are digital songs, games, bells, etc., whose price is limited (usually some dollars). For example, with reference to eBay, it should be hard to compute v_m since the range of possible values for transactions in eBay is really very large, and in any case such a normalization operation would have no sense.

Finally, we have the parameter Collusive $C = \left(\frac{1}{T_{i,j}}\right)^e$ where $T_{i,j}$ is the number of previous transactions between U_i and U_j, and e is a parameter, called *feedback relevance*, having a non-negative value and used to contrast possible collusive user behaviors. Observe that C is 1 (thus, it does not decrease α) only the first time U_i performs a transaction with U_j and becomes more and more (in function of the value of e) close to 0 when $T_{i,j}$ raises. Thus e decreases the final value of α and the relevance of the feedback value in computing R_j. This allows us to reduce the impact of collusive user behaviors aiming at increasing or decreasing the reputation of U_j by providing a number of (positive or negative) feedbacks.

The parameter e has to be set in order to fix the maximum number of feedbacks (provided by the same user to U_j) that has to be considered in computing R_j. Figure 2 shows how to set e. In particular, assuming that a value less than y can be considered negligible and that we want to consider significant only the first x transactions between the same users, the e-curve intersecting the point of coordinate x and y gives us the value of e. For example, chosen $x = 10$ and $y = 10^{-2}$, we have to fix $e = 2$. Thus, only the first 10 positive (resp. negative) transactions between U_i and U_j have a (progressively decreasing) significance in computing the new rating when the feedback is positive (resp. negative). The subsequent feedbacks between the two users give a negligible contribution (less than 0.01).

We show now an example of the importance of the parameter e in contrasting collusive behaviors. In Figure 3 we plot the reputation value of a user U during his first 25 transactions, using different values of e to compute his new reputation after every transaction. In order to amplify the effects of the different values of e, we consider an ad-hoc scenario where (1) both the reputation of the other users and the transactions have a high value (this produces rapid variations of the reputation of U) and (2) the overall number of users is very limited. Looking at Figure 3, we observe that in the first 9 transactions U receives a positive feedback, so that his reputation raises. Moreover, since the curves for all values of e are overlapped, we guess that these transactions are performed by U with different users and that the term Collusive of the reputation model has always been 1. Then, U receives at the following three transactions (transactions 10, 11 and 12) a negative feedback by the same user. At transaction 10, since it is their

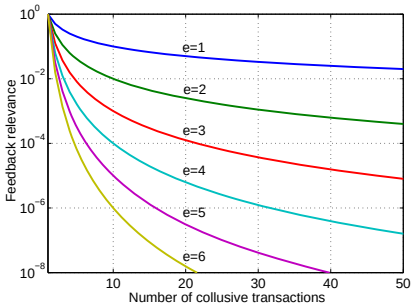


Fig. 2 Setting of the parameter e.

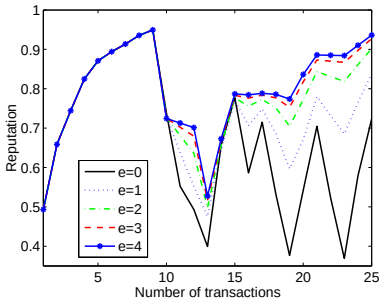


Fig. 3 Impact of e on collusive behaviors.

first transaction, the curves continue to be overlapped, but at transaction 11 and 12, the negative feedback assumes a different weight for a different e. For example, for $e = 6$, the second negative feedback has a weight $1/2^6$, thus contrasting a possible hostile behavior aiming at decreasing U's reputation and smoothing strongly the fast variation of reputation of the curve for $e = 0$. The case for $e = 1, .., 4$ produces an intermediate behavior. This trend of the curves is confirmed also for the following transactions.

We conclude this section by discussing about the last parameter N of our reputation model. Consider another transaction started by U_i and U_j which is not concluded for some reason (for example, during file transfer the distance between the two cell phones has become higher than the operating range of Bluetooth causing the disconnection and the transaction failure). As remarked in the previous section, our reputation system is designed to penalize both users U_i and U_j to avoid that intentionally a user could trouble trading of other users. The parameter N introduced in Section 3.1 is used to decrease the user's reputation in case of transaction abort. Observe that in many cases it is impossible to establish who of the two users is responsible for the failure and then who has to be penalized. For this reason, we exploit the following empirical criterion. We update U_j's reputation as follows (here we use the same notation as above):

$$R_j = N \cdot \widetilde{R_j} \quad \text{where} \quad N = \left(1 - \frac{AT_j}{T_j+1}\right) \cdot \frac{AP_j}{AP_j+AP_i} \quad \text{and} \quad AP = \frac{AT}{T}$$

We explain now the meaning of the above expressions. Since we cannot establish who is responsible for the failure, for each user we define an *abort probability* (AP) as the number of aborted transactions (AT) he was involved in divided by the overall number of (aborted or concluded) transactions he has done (T). AP_j and AP_i are the abort probabilities of U_j and U_i respectively. AP allows us to estimate user's responsibility for the transaction abort. For example, in case the two users have the same T and AT, their responsibility is equally divided since $AP = 0.5$. Beside the responsibility, U_j is penalized also if the number of failed transactions (AT_j) he was

involved in is not negligible with respect to the overall number of transaction (T_j) plus 1. Note that the term +1 has been added to avoid that user's reputation becomes zero in case his first transaction fails. In summary, in case of abort the reputation of a user is decreased proportionally to (1) the number of times he has been involved in aborted transactions and (2) his responsibility for the abort. Clearly also the reputation of U_i is updated according to the expression above.

4 Cost Evaluation

In this section we discuss the cost of our approach in terms of amount of storage, communication, and computation from both user-side and server-side.

Let us start the discussion with the analysis of these aspects from the user's side. As we have seen in Section 2, each user has to store the public key of the Reputation Manager, his secret key, his hazard threshold and his credential, containing the public key, the ID, and the reputation value of the user, as well as the expiration data and the credential signature. Assuming that keys and signatures have size 128 bytes and that 8 bytes are sufficient for each ID, threshold, reputation value and expiration data (the sizes here given are the most frequent values for this type of information), user's cell phone must store a very limited amount of data, less than one kilobyte, to implement our proposal.

Consider now the communication cost for each transaction. Each user, after the Affiliation step, carries out a transaction with another user by completing the phases Presentation and Transaction. Note that since Affiliation is done *una tantum*, thus it does not depend on the number of users or transactions done, the cost of this operation can be neglected. Presentation requires the exchange of the credentials, and during the Transaction step users exchange messages containing the resource request/response and the signature of the transaction descriptor and, finally, they exchange the ciphered resource, the payment, and the cryptographic key. Among the several information exchanged above, surely the one with a larger size is the ciphered resource (typically its size is some megabyte). For this reason we can neglect the costs, in terms of communication, due to the resource request/response (usually some kilobyte), the signature of transaction descriptor (128 bytes), and the payment code and the key (less than one kilobyte), and we can consider solely the costs necessary for resource sending. As a consequence, the communication overhead required by our approach is negligible with respect to the cost of communication necessary for the transfer of the resource.

Regarding the computational cost, each transaction requires the cell phone to produce the signature of the transaction descriptor and the ciphering (in case of the vendor) or the deciphering (for the buyer) of the resource by a symmetric-key cryptographic algorithm. As remarked in [44],

we do not have to worry about the computational cost of such operations since nowadays many commercial products are equipped with cryptographic hardware accelerators to enhance the performance of security computations.

Now we discuss the costs from the side of the Reputation Manager. Firstly, we consider the storage space necessary to manage users' reputation. As described in Section 3.2, our model requires to store for each user his credential and some other data necessary to update his reputation. Such data are the number of positive and negative feedbacks he has received, the number of aborted transactions he has effected, and the list of the number of transactions done with each other member. Thus, the cost in terms of storage space is proportional to $N_U \cdot N_T$, where N_U is the number of users of the system and N_T is the average number of users with whom each member has effected transactions. Since the same spatial cost is required by eBay [21], that is successfully deployed, we realize that the storage space necessary to implement our approach is acceptable.

From the computational and communication point of view, the Reputation Manager has to (1) receive and validate the signature of feedbacks provided by the member, (2) compute the new reputation for each user according to the reputation equation and the received feedbacks, and, then, (3) sign the new credential to send to the user. Since the cost of such operations is linear in the number of users, it is only necessary to upgrade the Reputation Manager as the number of users increases. Finally, concerning the communication between users and Reputation Manager, observe that it regards only the sending of feedbacks and the new credential, that, as remarked above, have a very limited size.

5 Related Work

The relevance of the reputation issues in many disciplines is testified by the rich literature that has produced various notions of reputation, each one with its own properties and models [11,12,16,31,43,47,50].

A first classification of reputation systems is based on the modality of users' reputation spread. In centralized systems a single unit receives information necessary to update users' reputations and makes users' reputations available to all members of the community. Differently, in decentralized systems reputation values have to be obtained by known users and/or neighbors, possibly propagating such reputation queries to neighbors of neighbors.

A well known centralized reputation system, based on feedbacks provided by members and a very simple metrics is exploited in eBay [21], where each user after a transaction receives +1, 0 or -1 as feedback and his reputation is computed as the sum of feedbacks obtained over the last six months. Many authors have deeply investigate this system and doubtless its main advantage is the simplicity. Conversely, it is sensitive to a lot of malicious behaviors, in particular collusive activities. Another simple centralized technique is adopted by Amazon [18]. As for eBay, also this approach does not

take into account some important elements like the context and the source of the information. This lack has been pointed out also in our experiments, where collusive activities have allowed malicious users to gain reputation. Another important centralized reputation system is SPORAS [59], that contrasts collusive alliances of users to increase reciprocally their reputations, by limiting the number of times an agent may increase the reputation of another agent. Moreover, in SPORAS a very low initial reputation rate is used to tackle the multiple identities problem. This technique has been considered in our experimental comparison. The results of the experiments have shown the superiority of CellTrust with respect to SPORAS in contrasting such types of malicious activities.

A reputation model thought for open multi-agent systems is FIRE [22]. In FIRE, four types of trust and reputation sources (resp., direct trust, role-based trust, witness reputation and certified reputation provided by always trusted witness) are introduced to cover the most usual circumstances. Such varieties of sources give FIRE a good versatility, but the drawback is that a lot of parameters need to be tuned for a correct working. However, the accuracy of a reputation model generally increases with the number of the subjects cooperating [6]. The cooperation can be reached following two approaches. In the first case, it is not compulsory to provide feedbacks (positive reputation systems), while in the second one (negative reputation systems) a penalty (resp., promotion) is introduced if feedbacks are not provided (resp., provided).

The aggregation of the reputation evaluations can be obtained using different rules and an interesting mechanism is proposed by REGRET [49], where the reputation is based on the aggregation, called social dimension, of the *impressions* about other agents (altruistic and cooperative) obtained by means of direct interactions. In REGRET each impression, called individual dimension, is composed by the relevance (weight) provided to the elements of a common semantic (ontological dimension) in accord with the personal point of view. By knowing the individual weights, it is possible to uniform each rate to the user's points of view. However, since the reputation depends on such weights, its value is subjective for each user and this makes this approach unusable in our scenario, where an absolute value of reputation for each user is spread by means of credentials.

An incentive for the development of new reputation models has been given by the diffusion of P2P networks and by the consequent need of reliability, in particular with respect to files and Web Service sharing [24,31, 57]. Many works have studied such environments and proposed a mix of different techniques. Usually, a great attention is given to the correctness of the reputation rates when malicious elements occur.

The EigenTrust Algorithm [25], inspired by the PageRank algorithm used by Google, adopts a distributed reputation system. It computes the reputation of each peer adding in a weighted manner all the reputation scores relative to satisfactory transactions, normalized over all the participants. EigenTrust assumes the transitivity of trustworthiness and each user

weighs rates received from other users by means of the trustworthiness that he gives to such witnesses. Global trust values are computed in distributed manner by updating trust values by means of the neighboring peers. The authors show that trust values, arranged in a trust matrix, asymptotically converge to the eigenvalue of such a matrix. This computation is conditioned by the presence of pre-trusted users, always trusted, that are used to minimize the influence of malicious elements in collusion activities. Such an approach cannot be exploited in our scenario both because it requires each user to contact many other users to compute the reputation, and because, as we have said for REGRET, the value of reputation is not absolute, so that it cannot be propagated by credentials. Moreover, it relies on pre-trusted users that can be difficultly guaranteed in our scenario. Another reputation model is PeerTrust [58], where a reputation-based trust framework is implemented in an adaptive manner and a PKI infrastructure is adopted to better security and reliability of the system. This paper is very interesting and our approach shares many characteristics with PeerTrust. However it is designed for stable P2P networks where by means of Distributed Hash Tables [2] it is possible to store across the network, in a distributed manner, trust data that are needed to compute the trust measure. For this reason this approach cannot be used in our case.

The proposal of NICE [53] consists in computing a trust rate based on both a local inference and a distributed search. Each transaction is annotated by the system in a cookie; then such cookies will be exploited to infer transitive trust along the users' chains. If we are in presence of uncertain, fuzzy, and incomplete information, the use of fuzzy techniques might be interesting. They are explored in [54], where the reputation system collects from each user his evaluations about the other users and weighs them in a suitable way in order to obtain an aggregated reputation score. Social networks and probabilistic models are examined in [13] along with different contexts and settings to determine the trustworthiness of the users. The authors conclude that in several scenarios the two techniques exhibit unsatisfactory performances.

Finally, some works have investigated trust and reputation issues in a mobile Ad Hoc network environment. In [42] several trust learning schemes based on both experience and recommendations, suitably adapted, allow information sources to specify their trust in the information provided by others. In [41] an implicit reputation management system is proposed. Each transaction is carried out only after a voting of the neighbors acting as witnesses. In this way, neighbors evaluate also the trustworthiness of the transaction actors based on their experience and the involved context. In accord to Ad Hoc network nature, there is a strong risk of obtaining only a highly partial depiction of user's reputation or that useful witnesses may be absent among the neighbors. A Bayesian method is adopted in [7], where the reputation is updated based on the "second-hand" criterion and the transitive reputation is accepted only if it is in accord with the direct rates. Instead, a reputation system based on a deviation test, independently of spe-

cific implementation assumptions, within a stochastic process is exploited in [36] to contrast liars in Ad Hoc networks; tests show that this model works well only until the number of liars do not exceed a certain threshold.

In summary, the most part of the cited works require to contact on-line other users or a central server in order to know the reputation of the counterpart. If such contacts can not be realized, these reputation systems result inoperative or, at best, can have only a partial representation of the reputation within a community. The remaining part of the cited work assumes that reputation data propagate among the users during their activities; also this approach permits only a partial knowledge of reputation data (with the exclusion of little communities of users in limited environments).

6 Experiments

In this section, we describe the experiments performed to test the efficacy of CellTrust in detecting malicious users in C2C trading activities. To this end, we have compared the performances of our proposal with two other reputation mechanisms, eBay [21] and SPORAS [59], already introduced in Section 5. The former is one of the most famous reputation systems developed and successfully deployed. Its reputation model is very trivial and considers the number of positive feedbacks received with respect to the overall number of transactions done. As remarked in Section 3.2, in eBay a member can increase or decrease another member's score only one time, no matter how many transactions they share.

A more sophisticated model is presented in SPORAS [59], a research proposal that, despite its non-young age, is still considered very effective and used as comparison model [8, 9, 22]. In SPORAS, the rating R_i (a value ranging from 0 to $D = 3000$) of a user after the i-th transaction is computed as follows:

$$R_i = R_{i-1} + \frac{1}{\theta} \cdot \Phi(R_{i-1}) R_i^{other} (W_i - E_i)$$

$$\Phi(R_{i-1}) = 1 - \frac{1}{1+e^{-(R_{i-1}-D)\sigma}} \qquad E_i = R_{i-1}/D$$

where θ is the effective number of ratings considered in the reputation evaluation, R_{i-1} is the old user's reputation, R_i^{other} is the reputation of the user giving the feedback W_i, and finally σ has been set to 0.11 according to the criterion given in [59]. Observe that, for comparison reasons, the reputation of the user has been normalized to the range from 0 to 1.

6.1 Parameters and Evaluation Metrics

Below, the parameters and the evaluation metrics considered in our experiments are described. They are:

- *Users population.* We considered a population of 1,000 users. This is the overall number of users who (potentially) can perform trading. Trading is performed between two, randomly chosen, users belonging to this population.
- *Number of malicious users.* This parameter varies from 0% to 95% of the overall number of users. The default value is 5%. We considered different malicious attacks that will be described in detail.
- *Number of transactions.* In our simulations we varied the number of transactions performed by each user. The default value is 10.
- *Transactions value.* The maximum value of a transaction has been fixed to 5 dollars, compatible with a real scenario for digital song trading, games for mobile, and so on. The transaction value distribution is modelled by a uniform distribution, thus all values between 1 and 5 (included) have the same probability.
- *Evaluation metrics.* We used the following three notions of error as evaluation metrics.
  - MEE (Malicious user Estimation Error) is computed as $\frac{\sum_{i=0}^{m} R(U_i)}{m}$ where m is the number of malicious users and $R(U_i)$ is the reputation of the i-th malicious user. In words, MEE is the average of the reputations of cheaters and defines an error since cheater's reputations should be 0 in a perfect reputation system.
  - HEE (Honest user Estimation Error) is computed as $\frac{\sum_{j=0}^{h}(1-R(U_j))}{h}$ where h is the number of honest users and $R(U_j)$ is the reputation of the j-th one. HEE is the error made by the reputation system in identifying honest users.
  - The last error GEE (Global Estimation Error) is the overall error in estimation of both bad and good users and is defined as $\frac{\sum_{i=0}^{m} R(U_i)+\sum_{j=0}^{h}(1-R(U_j))}{m+h}$.

  Clearly, the lower these errors (MEE, HEE, and GEE), the better the accuracy of the technique.

6.2 Experimental Results

In this section we test the efficacy of the three compared reputation models to detect malicious users. In the first experiment we analyze the accuracy of the techniques as the number of cheaters increases. We assume that each user has not done any transaction before (this is an important parameter for eBay and SPORAS) and thus he has an initial reputation 0.5 (this value is significant only in our approach). We denote this initial setting as *fixed* initial conditions. We measure Global Estimation Error of the techniques after each user has done about 10 transactions versus the percentage of the increasing number of cheaters. The result of this experiment is reported in Figure 4. Not surprisingly, the best technique is eBay that always reports an error 0 (its line overlaps X axis). This is due to the particular and unreal setting of this first experiment and to the formula of eBay's reputation

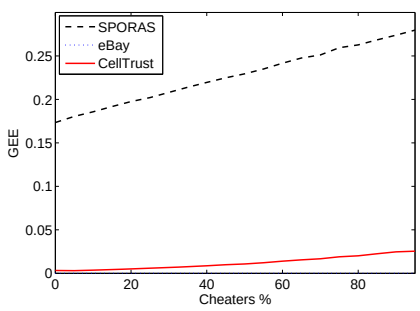


Fig. 4 Fixed Initial Conditions.

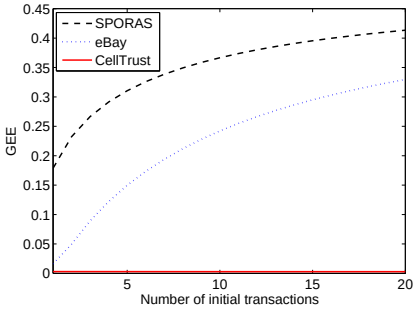
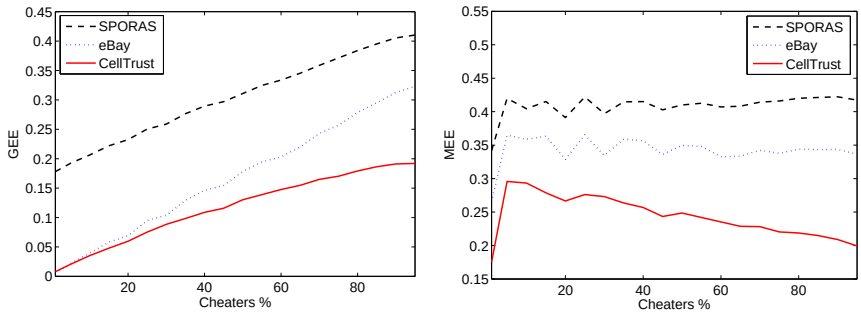


Fig. 5 Random Initial Conditions.

mechanism. In fact, since a malicious user misbehaves in all transactions, at the first bad transaction he does, eBay marks him as a cheater and will detect him as a malicious user in the future. However, as we will see in the following, whenever a user alternates bad and good actions, eBay makes wrong estimations. This experiment shows also that CellTrust accuracy is quite high (the error measured is always less than 0.025), specially when the percent of malicious users is low. On the contrary, SPORAS error is relevant and ranges from 0.17 to 0.25.

Now we repeat the previous experiment fixing the number of cheaters to 5% of the population and changing the initial conditions. In particular, we allow each user to do a given number of initial transactions in which his (bad or good) behavior is random. The error of the techniques is measured again after each user has done about 10 transactions in addition to the initial ones. The result of this experiment is reported in Figure 5 and points out that eBay dramatically has reduced its performance. The *noise* introduced by the initial transactions in which users behave randomly does not allow eBay to be as accurate as in an ideal situation where all users always act honestly or badly. Comparing the results depicted in Figures 4 and 5, we note that this noise does not disturb CellTrust (GEE maintains to 0.01) and affects partially SPORAS. In words, this experiment shows that the latter two techniques are more versatile than eBay to classify users when their behavior alternates between honest acting and cheating, as in real situations. In order not to penalize eBay, the following experiments will be done by setting fixed initial conditions, that are most favorable to eBay.

Now we analyze the accuracy of the three techniques when malicious users act honestly for low-value transactions and cheat whenever the value of the transaction is higher than a fixed threshold. The results here shown have been obtained setting this threshold to the value 2 dollars but we have measured similar results also when the threshold has been fixed to 3 and 4 (we recall that we have assumed that the maximum value of a transaction is 5 dollars). In Figure 6.(a) we depict GEE versus the number of cheaters. Observe that, when the number of cheaters is small, eBay and CellTrust

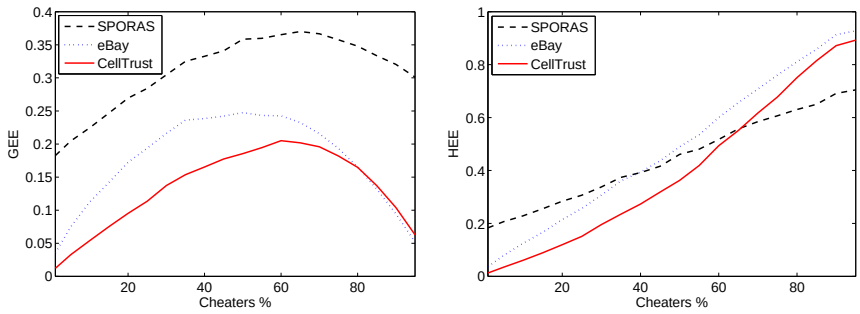


(a) Global Estimation Error (b) Malicious User Estimation Error

Fig. 6 Cheating based on price.

produce the same error (very close to zero) since the number of anomalous actions is negligible with respect to that of correct actions. When cheater percent raises, clearly all the techniques suffer from cheater's strategy and decrease their accuracy. Note that, among the methods, eBay has the worst performances (its range of error goes from 0 to 0.32, whereas CellTrust produces an error from 0 to 0.19 and SPORAS from 0.17 to 0.41). Another result of this experiment is shown in Figure 6.(b), where we report the component of Global Estimation Error due to the error in estimation of only cheaters. Clearly the weight of this component in computing GEE is proportional to the cheaters percentage. For example, when cheaters are 5 percent, MEE for CellTrust is 0.29 but its contribution to GEE is limited so that GEE keeps quite small (about 0.02). The use of MEE helps us to understand how much each method is penalized from this cheater strategy. CellTrust is much more able than the other techniques to contrast such an attack in which malicious user behavior is driven by the value of the transaction. We remark that such type of attack is currently and profitably done by some eBay members who sell and buy for free dummy resources to obtain positive feedbacks and thus increase their reputation.

In the next experiment we consider another type of malicious behavior, in which cheaters provide false negative feedbacks to discredit other users. The GEE error measured in this experiment is depicted in Figure 7.(a). The error lines of all methods have a particular trend: initially the error raises until cheaters are about 60 percent and then decreases. We motivate this trend by considering that when cheater percentage is low, we have a lot of honest users whose reputation is only partially injured by the few cheaters (then the summation in GEE is composed of a large number of little errors), whereas when cheaters are numerous, they can damage dramatically the small number of honest users, but also in this case GEE is limited because only few terms of GEE summation are high errors. Figure 7.(b), that plots HEE versus cheater percentage, confirms this conclusion. Here, we see that whenever the percent of cheaters is less than 60%, CellTrust is



(a) Global Estimation Error (b) Malicious User Estimation Error

Fig. 7 Malicious feedbacks.

the best method able to distinguish false negative feedbacks, whereas for higher percents, SPORAS shows the best accuracy, even if the error is very high (more than 0.5). However observe that in a realistic scenario we can assume that cheater percentage is very limited, so that CellTrust results the best candidate to contrast this type of attack.

We discuss now another possible threat carried out by malicious users who collude by exchanging positive feedbacks to gain a high reputation. The MEE error measured for the three methods is reported in Figure 8. When the number of cheaters is less than 35% of the population, CellTrust is the technique that guarantees the best accuracy. The feedback-credibility parameter of CellTrust allows us to give low relevance to feedbacks provided by low-reputation users. However, when the percentage of cheaters raises, the large number of malicious users providing false positive feedbacks allows cheaters to deceive CellTrust and to gain reputation. Observe also that eBay does not contrast this attack and malicious reputation raises linearly with the number of cheaters, whereas again SPORAS shows the best accuracy when the number of malicious users is unrealistically very high.

In conclusion, we summarize the results as follows. eBay is recommendable in completely trustworthy situations, where there are no cheaters, or whenever malicious user actions are "constant", that is, they always cheat. Its simple reputation model can guarantee good accuracy and high scalability. On the other hand, as we have shown in these experiments, it suffers from alternate behavior and from many attacks allowing cheaters to gain reputation and to defraud honest users. SPORAS is a more complex approach and has the best performance in highly untrustworthy situations where the number of cheaters is relevant with respect to the overall number of users. In this scenario, SPORAS can contrast many attacks implementable by malicious users.

Our proposal is a good compromise between the two techniques. Indeed, it guarantees the best accuracy in scenarios where the number of potential cheaters is small with respect to the overall number of users, that is the most

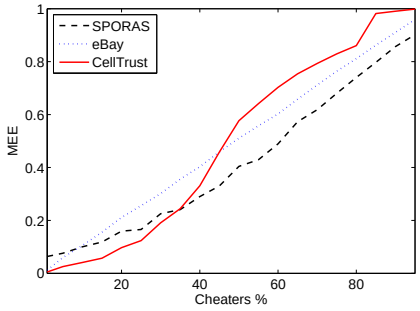


Fig. 8 Collusive actions.

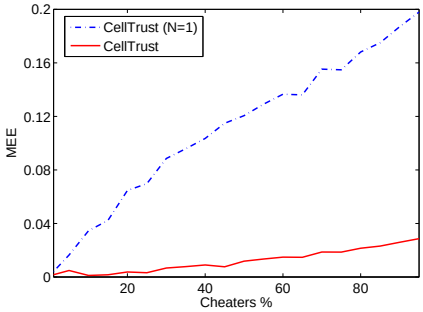


Fig. 9 Trading disturbing activity.

common scenario we find in real environments. Moreover, this reputation mechanism allows us to contrast the most frequent actions done by malicious users to gain reputation or to discredit honest users.

We conclude this section analyzing the efficacy of the parameter N, introduced in Section 3.2, used to penalize user activity aimed to disturb trading. In this experiment, malicious user activity is realized by starting a trading and then aborting it before the transaction is complete. We measured MEE error produced by our proposal versus the number of malicious users in two cases: when the parameter N is enable, that is, it is correctly computed according to the reputation metrics, and when N is "disabled" (this corresponds to set $N = 1$). The results of this experiment are shown in Figure 9 and confirm that the use of N improves the accuracy of CellTrust whenever malicious users annoy honest users starting unfruitful transactions. Note that the error measured for eBay and SPORAS, that is always higher than that of CellTrust, is not reported since these techniques are not designed to take into account such a malicious activity.

7 Conclusion

In this paper we have presented a novel reputation system designed to support C2C commerce activities done by users by means of their cell phones exploiting wireless technologies. Reputation is based on several parameters - like user's trading history, transaction value, feedbacks received - that allow us to contrast many usual attacks done by cheaters to gain reputation, realized for example, by providing false feedbacks, by collusion, by alternate behavior based on transaction value. Reputation spreading is based on the use of temporary signed credentials, that guarantee members' identity and reputation, issued by a centralized server that is also responsible for reputation computing. We have experimentally compared our proposal with two relevant reputation systems. The experiments have proved the effectiveness of our proposal in detecting malicious users in the most common situations.

References

1. Abdul-Rahman, A., Hailes, S.: Supporting Trust in Virtual Communities. In: HICSS '00: Proc. of the 33rd Hawaii Int. Conf. on System Sciences-Volume 6., vol. 6. IEEE Computer Society., Washington, DC, USA (2000)
2. Aberer, K.: P-grid: A Self-Organizing Access Structure for P2P Information Retrieval. In: COOPIS 2001: Proc. of the Ninth Int. Conf. on Cooperative Infor. Systems, *Lecture Notes in Computer Science*, vol. 2172, pp. 179–194. Springer-Verlag, London, UK. (2001)
3. Arboit, G., Crépeau, C., Davis, C.R., Maheswaran, M.: A localized certificate revocation scheme for mobile ad hoc networks. Ad Hoc Netw. **6**(1), 17–31 (2008)
4. Avancha, S., D'Souza, P., Perich, F., Joshi, A., Yesha, Y.: P2P M-Commerce in Pervasive Environments. SIGecom Exch. **3**(4), 1–9 (2003)
5. Banks, M.: The eBay Survival Guide: How To Make Money and Avoid Losing Your Shirt. No Starch Press, San Francisco, CA, USA (2005)
6. Birk, A.: Boosting Cooperation by Evolving Trust. Applied Artificial Intelligence. **14**(8), 769–784 (2000)
7. Buchegger, S., Le Boudec, J.Y.: A Robust Reputation System for P2P and Mobile Ad-hoc Networks. In: P2PEcon 2004: Proc. of the 2nd Workshop on the Economics of Peer-to-Peer Systems. Harward, MA, USA (2004)
8. Carbó, J., Molina, J.M., Dávila, J.: Trust Management Through Fuzzy Reputation. Journal of Cooperative Information Systems (1), 135–155 (2003)
9. Carbó, J., Molina, J.M., Dávila, J.: Fuzzy Referral Based Cooperation in Social Networks of Agents. AI Commun. **18**(1), 1–13 (2005)
10. Chinni, S., Thomas, J., Ghinea, G., Shen, Z.: Trust model for certificate revocation in ad hoc networks. Ad Hoc Netw. **6**(3), 441–457 (2008)
11. Conte, R., Paolucci, M.: Reputation in Artificial Societies: Social Beliefs for Social Order. Kluwer Academic Publishers, Hingham, MA, USA. (2002)
12. Dellarocas, C.: The Digitization of Word of Mouth: Promise and Challenges of Online Feedback Mechanisms. Manage. Sci. **49**(10), 1407–1424 (2003)
13. Despotovic, Z., Aberer, K.: P2P Reputation Management: Probabilistic Estimation vs. Social Networks. Comput. Networks. **50**(4), 485–500 (2006)
14. Falcone, R., Castelfranchi, C.: Social Trust: a Cognitive Approach. Kluwer Academic Publishers, Norwell, MA, USA. (2001)
15. Falcone, R., Castelfranchi, C.: The Socio-cognitive Dynamics of Trust: Does Trust Create Trust?. In: Proc. of the Workshop on Deception, Fraud, and Trust in Agent Societies Held During the Autonomous Agents Conf., pp. 55–72. Springer-Verlag, London, UK. (2001)
16. Grandison, T., Sloman, M.: A Survey of Trust in Internet Application. IEEE Communication Surveys. Fourth Quarter (2000)
17. Housley, R., Polk, W., Ford, W., Solo, D.: RFC 3280 - Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile (2002). http://www.faqs.org/rfcs/rfc3280.html
18. http://auctions.amazon.com : 2008
19. http://grouper.ieee.org/groups/802/11/index.html : 2008
20. http://www.bluetooth.com : 2008
21. http://www.ebay.com : 2008
22. Huynh, T.D., Jennings, N.R., Shadbolt, N.R.: An Integrated Trust and Reputation Model for Open Multi-Agent System. Autonmous Agent and Multi Agent Systems **13**, 119–154 (2006)

23. Josang, A., Gray, E., Kinateder, M.: Simplification and Analysis of Transitive Trust Networks. Web Intelli. and Agent Sys. **4**(2), 139–161 (2006)
24. Josang, A., Ismail, R., Boyd, C.: A Survey of Trust and Reputation Systems for Online Service Provision. Decision Support System **43**(2), 618–644 (2005)
25. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigentrust Algorithm for Reputation Management in P2P Networks. In: WWW '03: Proc. of the 12th International Conf. on World Wide Web., pp. 640–651. ACM Press, New York, NY, USA. (2003)
26. Karp, D.A.: eBay Hacks : 100 Industrial-Strength Tips & Tools. O'Reilly Media, Inc., San Francisco, CA, USA (2003)
27. Kleist, V.F.: A Transaction Cost Model of Electronic Trust: Transactional Return, Incentives for Network Security and Optimal Risk in the Digital Economy. Electronic Commerce Research. **4**(1-2), 41–57 (2004)
28. Kong, J., Luo, H., Xu, K., Gu, D.L., Gerla, M., Lu, S.: Adaptive Security for Multi-layer ad-hoc networks. Wireless Communication and Mobile Computing **2**, 533–547 (2002)
29. Lax, G., Sarné, G.M.L.: Blue: a Reputation-based Multi-Agent System to Support C2C in P2P Bluetooth Networks. In: ICE-B 2006: Proc. of the International Conference on E-Business, pp. 97–104 (2006)
30. Marsh, S.P.: Formalizing Trust as a Computational Concept, PhD Thesis, Department of Computing Science and Mathematics, University of Stirling. (1994)
31. Marti, S., Garcia-Molina, H.: Taxonomy of Trust: Categorizing P2P Reputation Systems. Comput. Networks. **50**(4), 472–484 (2006)
32. Massa, P.: A Survey of Trust Use and Modeling in Current Real Systems. In: R. Song, L. Korba, G. Yee (eds.) Trust in E-Services: Technologies, Practices and Challenges. Idea Group Publishing (2006)
33. Misra, S.K., Wickamasinghe, N.: Security of a Mobile Transaction: A Trust Model. Electronic Commerce Research. **4**(4), 359–372 (2004)
34. Misztal, B.M.: Trust in Modern Societies. Polity, Cambridge, England. (1996)
35. Mui, L., Mohtashemi, M., Halberstadt, A.: Notions of Reputation in Multi-Agents Systems: a Review. In: Proc. of the First Int. Joint Conf. on Autonomous Agents and Multiagent Systems (AAMAS '02), pp. 280–287. ACM Press, New York, NY, USA. (2002)
36. Mundinger, J., Le Boudec, J.Y.: The Impact of Liars on Reputation in Social Networks. In: Proc. of Social Network Analysis: Advances and Empirical Application Forum (2005)
37. Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C.: RFC 2560 - X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP (1999). http://www.faqs.org/rfcs/rfc2560.html
38. O'Mahony, D., Pierce, M., Tewari, H.: Electronic Payment Systems for E-Commerce. Artech House, Inc., Norwood, MA, USA. (2001)
39. Ondrus, J., Pigneur, Y.: Towards a Holistic Analysis of Mobile Payments: A Multiple Perspectives Approach. Electronic Commerce Research and Applications. **5**(3), 246–257 (2006)
40. Patton, M.A., Josang, A.: Technologies for Trust in Electronic Commerce. Electronic Commerce Research. **4**(1-2), 9–21 (2004)
41. Perich, F., Joshi, A., Yesha, Y., Finin, T.: Neighborhood-Consistent Transaction Management for Pervasive Computing Environments. In: Proc. of the 14th International Conf. on Database and Expert Systems Applications

(DEXA 2003), *Lecture Notes in Computer Science*, vol. 2736, pp. 276–286. Springer (2003)
42. Perich, F., Undercoffer, J., Kagal, L., Joshi, A., Finin, T., Yesha, Y.: In Reputation We Believe: Query Processing in Mobile Ad-hoc Networks. In: Proc. of the First Annual International Conf. on Mobile and Ubiquitous Systems: Networking and Services, 2004 (MOBIQUITOUS 2004), pp. 326–334. Maryland Univ., Baltimore, MD, USA. (2004)
43. Ramchurn, S.D., Huynh, D., Jennings, N.R.: Trust in Multi-Agent Systems. Knowl. Eng. Rev. **19**(1), 1–25 (2004)
44. Ravi, S., Raghunathan, A., Kocher, P., Hattangady, S.: Security in Embedded Systems: Design Challenges. Trans. on Embedded Computing Sys. **3**(3), 461–491 (2004)
45. Reichling, F.: Effects of Reputation Mechanisms on Fraud. Prevention in eBay Auctions, Thesis, Stanford University. (2004)
46. Resnick, P., Zeckhauser, R.: Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. In: M.R. Baye (ed.) The Economics of the Internet and E-Commerce, *Advances in Applied Microeconomics*, vol. 11. Elsevier Science (2002)
47. Resnick, P., Zeckhauser, R., Friedman, E., Kuwabara, K.: Reputation Systems. Commununication of ACM. **43**(12), 45–48 (2000)
48. Rietjens, B.: Trust and Reputation on eBay: Towards a legal Framework for Feedback Intermediaries. Information & Communications Technology Law. **15**(1), 55–78 (2006)
49. Sabater, J., Sierra, C.: REGRET: Reputation in Gregarious Societies. In: AGENTS '01: Proc. of the Fifth Int. Conf. on Autonomous Agents, pp. 194–195. ACM Press, New York, NY, USA. (2001)
50. Sabater, J., Sierra, C.: Review on Computational Trust and Reputation Models. Artificial Intelligence Review **24**(1), 33–60 (2005)
51. Serrano-Alvarado, P., Roncancio, C., Adiba, M.: A Survey of Mobile Transactions. Distrib. Parallel Databases. **16**(2), 193–230 (2004)
52. Shaked, Y., Wool, A.: Cracking the Bluetooth PIN. In: Proc. of the 3rd Int. Conf. on Mobile Systems, Applications, and Services (MobiSys '05), pp. 39–50. ACM Press, New York, NY, USA. (2005)
53. Sherwood, R., Lee, S., Bhattacharjee, B.: Cooperative Peer Groups in NICE. Comput. Networks. **50**(4), 523–544 (2006)
54. Song, S., Hwang, K., Zhou, R., Kwok, Y.K.: Trusted P2P Transactions with Fuzzy Reputation Aggregation. IEEE Internet Computing. **9**(6), 24–34 (2005)
55. Tan, Y.H., Thoen, W.: An Outline of a Trust Model for Electronic Commerce. Applied Artificial Intelligence **14**(8), 849–862 (2000)
56. Veijalainen, J., Terziyan, V.Y., Tirri, H.: Transaction Management for M-Commerce at a Mobile Terminal. Electronic Commerce Research and Applications **5**(3), 229–245 (2006)
57. Wang, Y.F., Hori, Y., Sakurai, K.: Characterizing Economic and Social Properties of Trust and Reputation Systems in P2P Environment. Journal of Computer Science and Tecnology **23**(1), 129–140 (2008)
58. Xiong, L., Liu, L.: PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. IEEE Transactions on Knowledge and Data Engineering. **16**(7), 843–857 (2004)
59. Zacharia, G., Maes, P.: Trust Management Through Reputation Mechanisms. Applied Artificial Intelligence. **14**(9), 881–907 (2000)