A Robust Reputation System for P2P and Mobile Ad-hoc Networks *

Sonja Buchegger †
EPFL-IC-LCA
CH-1015 Lausanne, Switzerland
sonja.buchegger@epfl.ch

Jean-Yves Le Boudec
EPFL-IC-LCA
CH-1015 Lausanne, Switzerland
jean-yves.leboudec@epfl.ch

Abstract

Reputation systems can be tricked by the spread of false reputation ratings, be it false accusations or false praise. Simple solutions such as exclusively relying on one's own direct observations have drawbacks, as they do not make use of all the information available. We propose a fully distributed reputation system that can cope with false disseminated information. In our approach, everyone maintains a reputation rating and a trust rating about everyone else that they care about. From time to time first-hand reputation information is exchanged with others; using a modified Bayesian approach we designed and present in this paper, only second-hand reputation information that is not incompatible with the current reputation rating is accepted. Thus, reputation ratings are slightly modified by accepted information. Trust ratings are updated based on the compatibility of second-hand reputation information with prior reputation ratings. Data is entirely distributed: someone's reputation and trust is the collection of ratings maintained by others. We enable redemption and prevent the sudden exploitation of good reputation built over time by introducing re-evaluation and reputation fading.

1 Introduction

1.1 Motivation

Reputation systems have been proposed for a variety of applications, among them are the selection of good peers in a peer-to-peer network, the choice of transaction partners for online auctioning such as E-bay [20], and the detection of misbehaving nodes in mobile ad-hoc networks [3]. There is a trade-off between efficiency in using the available information and robustness against false ratings [4]. If the ratings made by others are considered, the reputation system can be vulnerable to false accusations or false praise. However, if only one's own experience is considered, the potential of learning from experience made by others goes unused. Using only positive or only negative information reduces the vulnerability to only false praise or only false accusations.

Our goal is to make existing systems both robust against false ratings and efficient at detecting misbehavior. We propose a mechanism that makes use of all the available information, i.e. both positive and negative, both from own and from others' experience. To make the reputation system robust we include a way of dealing with false ratings.

In the remainder of this paper we refer to the entities in the reputation system as nodes, since we apply the reputation system to peer-to-peer and mobile ad-hoc networks.

1.2 Solution Overview

The main properties of a reputation system are the representation of reputation, how the reputation is built and updated, and for the latter, how the ratings of others are considered and integrated. The reputation of a given node is the collection of ratings maintained by others about this node. In our approach, a node i maintains two ratings about every other node j that is cares about. The *reputation rating* represents the opinion formed by node i about node j's behavior as an actor in the base system (for example, whether node j correctly participates in the routing protocol of a mobile ad-hoc network, or whether it provides correct files in a peer-to-peer file-sharing system). The *trust rating* represents node i 's opinion about how honest node j is as an actor in the reputation system (i.e. whether the reported first hand information summaries published by node j are likely to be true). We represent the ratings that node i has about node j as data structures $R_{i,j}$ for reputation and $T_{i,j}$ for trust. In addition, node i maintains a summary record of *first hand information* about node j in a data structure called $F_{i,j}$.

To take advantage of disseminated reputation information, i.e., to learn from observations made by others before having to learn by own experience, we need a means of incorporating the reputation ratings into the views of others. We do this as follows. First, whenever node i makes a first hand observation of node j's behavior, the first hand information $F_{i,j}$ and the reputation rating $R_{i,j}$ are updated. Second, from time to time, nodes publish their first-hand information to a subset of the population. Say that node i receives from k some first hand information $F_{k,j}$ about node j. If k is classified as "trustworthy" by i, or if $F_{k,j}$ is close to $R_{i,j}$ (in a sense that is made precise in Section 3.3) then $F_{k,j}$ is accepted by i and is used to slightly modify the rating $R_{i,j}$. Else, the reputation rating

* In Proceedings of P2PEcon 2004, Harvard University, Cambridge MA, U.S.A., June 2004

† The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

is not updated. In all cases, the trust rating $T_{i,k}$ is updated; if $F_{k,j}$ is close to $R_{i,j}$, the trust rating $T_{i,k}$ slightly improves, else it slightly worsens. The updates are based on a modified Bayesian approach we designed and present in this paper, and on a linear model merging heuristic.

Note that, with our method, only first hand information $F_{i,j}$ is published; the reputation and trust ratings $R_{i,j}$ and $T_{i,j}$ are never disseminated.

The ratings are used to make decisions about other nodes, which is the ultimate goal of the entire reputation system. For example, in a mobile ad-hoc network, decisions are about whether to forward for another node, which path to choose, whether to avoid another node and delete it from the path cache, and whether to warn others about another node. In our framework, this is done as follows. Every node uses its rating to periodically classify other nodes, according to two criteria: (1) normal/misbehaving (2) trustworthy/untrustworthy. Both classifications are performed using the Bayesian approach, based on reputation ratings for the former, trust ratings for the latter.

2 Related Work

False accusations are not an issue in positive reputation systems, since no negative information is kept [16, 7], however, the disseminated information could still be false praise and result in a good reputation for misbehaving nodes. Moreover, even if the disseminated information is correct, one cannot distinguish between a misbehaving node and a new node that just joined. Many reputation systems build on positive reputation only [21], some couple privileges to accumulated good reputation, e.g. for exchange of gaming items or auctioning [20]. Josang and Ismail [13] also use a Bayesian approach, it is, however, centralized and discounts the belief in second-hand information according to the reputation of agents, equating the trustworthiness of an agent as a witness with its performance in the base system. Lying nodes that aim at changing the reputation of another node, however, can still perform normally in the base system. We therefore separate the performance in the base system (reputation) from the one in the reputation system itself (trust). Mui et al. [17] present a computational model for reputation which is also based on Bayesian estimation, but uses chains of reputations ratings to obtain indirect measures of reputation, assuming trust transitivity. Another centralized reputation system exploiting reputation paths between entities has been proposed by Zacharia et al. [22].

A reputation-based trust management has been introduced by Aberer and Despotovic in the context of peer-to-peer systems [1], using the data provided by a decentralized storage method (P-Grid) as a basis for a data-mining analysis to assess the probability that an agent will cheat in the future given the information of past transactions. The disseminated information is exclusively negative, in the form of complaints that are then redundantly stored at different agents. When agents want to assess the trustworthiness of other agents, they query several agents for complaints about the agent in question. To assess the trustworthiness of the agents responding to the query , a complaint query about the complaining agent can be made, and so on.

A formal model for trust in dynamic networks based on intervals and a policy language has been proposed by Carbone et al. [5]. They express both trust and the uncertainty of it as trust ordering and information ordering, respectively. They consider the delegation of trust to other principals. In their model, only positive information influences trust, such that the information ordering and the trust ordering can differ. In our system, both positive and negative information influence trust and certainty, since we prefer p positive observations that come out of n total observations to p out of N when $n < N$.

Collaboration enforcement for peer-to-peer networks have been proposed by Moreton and Twigg [19]. They allow for selective trust transitivity and distinguish between trust as participator and trust as recommender. They define three operators, namely discounting, consensus, and difference, to compute trust values. Since they use recommenders, trust in participators, trust in recommenders, and meta-recommenders, the trust becomes recursive and they thus look for fixed-point solutions to the resulting trust equations.

Given that there is no incentive for truthful reputation reporting because reporting provides a competitive advantage to others, reporting positive ratings lead to a relative degradation of the reputation of the reporter, and by reporting fake negative ratings it is improved. Jurca and Faltings [14] aim for an incentive-compatible mechanism by introducing payment for reputation. The agents pay to receive reputation ratings from so-called R-agents, which in turn pay agents providing the information. Agents only get payed for their reputation report if the next agent report has the same result. This approach is interesting in that contains a heuristic to address dishonest ratings, which however does not take into account collusion, where nodes could have influence on the next report. To encourage the exchange of reputation information, Pinocchio [10] rewards participants that advertise their experience to others and uses a probabilistic honesty metric to detect dishonest users and deprive them of the rewards. The reward can be used to query the reputation of others. Pinocchio does not intend to protect against conspiracies or bad-mouthing.

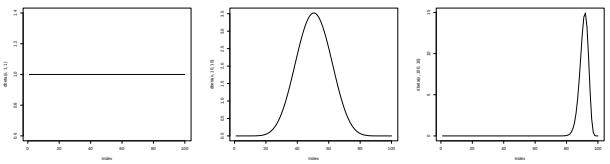
The EigenTrust mechanism [15] aggregates trust information from peer by having them perform a distributed trust calculation approaching the Eigenvalue of the trust matrix over the peers. The algorithm relies on the presence of pre-trusted peers, that is some peers have to be trusted, regardless their performance, prior to having interacted with them. The system relieves peer with bad performance from delivering files, due to their bad reputation. By isolating peers with bad reputation, the number of inauthentic downloads is decreased, however, if the motivation for misbehavior is selfishness, the misbehaved peers are rewarded.

3 Solution Proposal: A Bayesian Approach to Reputation Systems

3.1 A Bayesian Framework

Node i models the behavior of node j as an actor in the base system as follows. Node i thinks that there is a parameter θ such that node j misbehaves with probability θ, and that the outcome is drawn independently from observation to observation (Node i thinks that there is a different parameter θ for every different node j, and every node i may believe in different parameters θ; thus θ should be indexed by i and j, but for brevity, we omit the indices here). The parameters θ are unknown, and node i models this uncertainty by assuming that θ itself is drawn according to a distribution (the "prior") that is updated as new observations become available. This is the standard Bayesian framework. We use for the prior the distribution Beta(α, β), as is commonly done [2, 6], since it is suitable for Bernoulli distributions and the conjugate is also a Beta distribution.

The standard Bayesian procedure is as follows. Initially, the prior is Beta$(1, 1)$, the uniform distribution on $[0, 1]$; this represents absence of information about which θ will be drawn. Then, when a new observation is made, say with s observed misbehaviors and f observed correct behaviors, the prior is updated according to $\alpha := \alpha + s$ and $\beta := \beta + f$. If θ, the true unknown value, is constant, then after a large number n of observations, $\alpha \sim n\theta$ (in expectation), $\beta \sim n(1-\theta)$ and Beta(α, β) becomes close to a Dirac at θ, as expected. The advantage of using the Beta function is that it only needs two parameters that are continuously updated as observations are made or reported. See Figure 1 (the actual calculation of the density has been carried out here for illustrative purpose only).



(a) Non-informative Prior Beta(1,1). (b) Beta(10,10). (c) Beta(100,10).

Figure 1: Density of the Beta Function.

We use a modification of the standard Bayesian method, one for reputation, and one for trust, as described next.

3.2 Modified Bayesian Approach for First-Hand Information

The first-hand information record $F_{i,j}$ mentioned in the introduction has the form (α, β). It represents the parameters of the Beta distribution assumed by node i in its Bayesian view of node j's behavior as an actor in the base system. Initially, it is set to $(1, 1)$.

The standard Bayesian method gives the same weight to each observation, regardless of its time of occurrence. We want to give less weight to evidence received in the past to allow for reputation fading. We therefore developed a modified Bayesian update approach by introducing a moving weighted average as follows. Assume i makes one individual observation about j; let $s = 1$ if this observation is qualified as misbehavior (by a system such as CONFIDANT), and $s = 0$ otherwise. The update is

$$\alpha := u\alpha + s \quad (1)$$

$$\beta := u\beta + (1 - s) \quad (2)$$

The weight u is a discount factor for past experiences, which serves as the fading mechanism.

We now analyze how to find a good value of u. Call $s_1, ..., s_n$ the sequence of observations. We can easily derive from Equation (1) that the value of α after n first hand observations is

$$\alpha_n = s_n + us_{n-1} + ... + u^{n-1}s_1 + u^n \quad (3)$$

Assume (temporarily) that θ would be constant. For large n we would have

$$\mathbb{E}(\alpha_n) \approx \frac{\theta}{1-u} \quad (4)$$

$$\mathbb{E}(\beta_n) \approx \frac{1-\theta}{1-u} \quad (5)$$

Assume in addition that $m = \frac{1}{1-u}$ is an integer. Thus the standard Bayesian approach after m observations would result in the same posterior as ours after infinitely many observations. Thus, as a rule of thumb, we should select u as

$$u = 1 - \frac{1}{m} \quad (6)$$

where m is the order of magnitude of the number of observations over which we believe it makes sense to assume stationary behavior.

In addition, during inactivity periods, we periodically decay the values of α, β as follows. Whenever the inactivity time expires, we let $\alpha := u\alpha$ and $\beta := u\beta$. This is to allow for redemption even in the absence of observations, either due retaliatory exclusion or simply lack of interaction.

3.3 Reputation Rating and Model Merging

The reputation rating $R_{i,j}$ is also defined by two numbers, say (α', β'). Initially, it is set to $(1, 1)$. It is updated on two types of events: (1) when first-hand observation is updated (2) when a reputation rating published by some other node is accepted and copied.

In the former case, the update is the same as for the first-hand information. More precisely: let $s \in \{0, 1\}$ be the observation:

$$\alpha' := u\alpha' + s \quad (7)$$

$$\beta' := u\beta' + (1 - s) \quad (8)$$

If the update to the first-hand information is due to inactivity, the formula is $\alpha' := u\alpha'$, $\beta' := u\beta'$.

In the latter case, we use linear pool model merging [2], as follows. Assume node i receives the reported first-hand information $F_{k,j}$ from node k. The question is how to detect and avoid false reports. Our approach is for a node i to take into account trust and compatibility. If $T_{i,k}$ is such that i considers k trustworthy according to Equation (14) (defined later), $F_{k,j}$ is considered by node i who modifies $R_{i,j}$ according to

$$R_{i,j} := R_{i,j} + wF_{k,j} \tag{9}$$

Here, w is a small positive constant [2]. This is performed for all j contained in the report.

Otherwise, i considers k untrustworthy, and, for every node j in the report, uses the results of the *deviation test*, as follows. We denote with $\mathbb{E}(\text{Beta}(\alpha, \beta))$ the expectation of the distribution $\text{Beta}(\alpha, \beta)$. Let $F_{k,j} = (\alpha_F, \beta_F)$ and $R_{i,j} = (\alpha, \beta)$ The deviation test is

$$|\mathbb{E}(\text{Beta}(\alpha_F, \beta_F)) - \mathbb{E}(\text{Beta}(\alpha, \beta))| \geq d \tag{10}$$

where d is a positive constant (deviation threshold). If the deviation test is positive, the first hand information $F_{k,j}$ is considered incompatible and is not used. Else $F_{k,j}$ is incorporated using Equation (9) as previously.

3.4 Trust Ratings

Trust rating uses a similar Bayesian approach. Node i thinks that there is a parameter ϕ such that node j gives false reports with probability ϕ, so it uses for ϕ the prior $\text{Beta}(\gamma, \delta)$. The trust rating $T_{i,j}$ is equal to (γ, δ).

Initially, $(\gamma, \delta) = (1, 1)$. Then an update is performed whenever node i receives a reported by some node k on first-hand information about node j. Let $s = 1$ if the deviation test in Equation (10) succeeds, and $s = 0$ otherwise. The trust rating $T_{i,k} = (\gamma, \delta)$ is updated by

$$\gamma := v\gamma + s \tag{11}$$

$$\delta := v\delta + (1 - s) \tag{12}$$

Here v is the discount factor for trust, similar to u. There is a similar update in periods of inactivity as for first hand information.

Note that the deviation test is always performed, whether k is considered trustworthy by i or not. In the former case, it is used only to update $T_{i,k}$; in the latter case, it is used to update $T_{i,k}$ and decide whether to update $R_{i,j}$.

3.5 Classification

The decision-making process works as follows. First, the posterior according to all the given data is calculated. This is done by node i by updating $R_{i,j} = (\alpha', \beta')$ and $T_{i,j} = (\gamma, \delta)$ as explained above. Then node i chooses the decision with minimal loss.

We use squared-error loss for the deviation from the true θ and ϕ; this amounts to considering $\mathbb{E}(\text{Beta}(\alpha', \beta'))$ for θ and $\mathbb{E}(\text{Beta}(\gamma, \delta))$ for ϕ. More precisely:

Node i classifies the behavior of node j as

$$\begin{cases} \text{normal} & \text{if } \mathbb{E}(\text{Beta}(\alpha', \beta')) < r \\ \text{misbehaving} & \text{if } \mathbb{E}(\text{Beta}(\alpha', \beta')) \geq r \end{cases} \tag{13}$$

and the trustworthiness of node j as

$$\begin{cases} \text{trustworthy} & \text{if } \mathbb{E}(\text{Beta}(\gamma, \delta)) < t \\ \text{untrustworthy} & \text{if } \mathbb{E}(\text{Beta}(\gamma, \delta)) \geq t \end{cases} \tag{14}$$

The thresholds r and t are an expression of tolerance. If node i tolerates a node j that misbehaves not more than half of the time, it should set r to 0.5. In analogy, if i trusts a node if its ratings deviate no more than in 25% of the cases, it sets its t to 0.75. In the case of an asymmetric loss function for false positives and negatives, the classification has to be modified.

3.6 Issues in Distributed Reputation Systems

Should liars be punished? If we punish nodes for their seemingly inaccurate testimonials, we might end up punishing the messenger and thus discourage honest reporting of observed misbehavior. Note that we evaluate testimonial accuracy according to affinity to the belief of the requesting node along with the overall belief of the network as gathered over time. The accuracy is not measured as compared to the actual true behavior of a node, since the latter is unknown and can not be proved beyond doubt. Even if it were possible to test a node and obtain a truthful verdict on its nature, a contradicting previous testimonial could still be accurate. Thus, instead of punishing deviating views we restrict our system to merely reduce their impact on public opinion. Some node is bound to be the first witness of a node misbehaving, thus starting to deviate from public opinion. Punishing this discovery would be counterproductive, as the goal is precisely to learn about misbehaving nodes even before having had to make a bad experience in direct encounter. Therefore, in our design, we do not punish a node when it is classified as untrustworthy.

Identity. The question of identity is central to reputation systems. We require three properties of identity which we call persistent, unique, and distinct. We are investigating the use of expensive pseudonyms [11], cryptographically generated unique identifiers [18], and secure hardware modules [12] to counter identity problems such as the Sybil attack [9].

Redemption. Our solution enforces redemption of nodes over time, by the combination of two mechanisms: periodic re-evaluation and reputation fading. Periodic re-evaluation is implemented by the fact that node classification is performed periodically. It is thus possible for a node to redeem itself, given that nodes have each their own reputation belief which is not necessarily shared by all the others. Since their opinions can differ, a node is most probably not excluded by all other nodes and can thus partially participate in the network with the potential of showing its good behavior. Even if this is not the case and the suspect is excluded by everyone it can redeem

itself by means of the second mechanism. Reputation fading is implemented by our modification to the Bayesian update of the posterior, which decays exponentially. Contrary to standard Bayesian estimation, this gives more weight to recent observations. We also periodically discount the rating in the absence of testimonials and observations.

4 Performance Evaluation

We evaluate the performance of our reputation system as applied to CONFIDANT [3], a misbehavior detection system for mobile ad-hoc networks, using the GloMoSim [23] simulator. We are interested in its performance according to the following metrics.

1. Detection time of misbehaving nodes. We measure this as the simulation time taken for all misbehaving nodes to be classified as detected by all normal nodes. Figure 2 shows the mean for a network of 50 nodes. To show the effect of presence or absence of the reputation system model merging, we set w, the weight for second-hand observations, to 0.1 and 0, the latter meaning that nodes do not consider second-hand information at all. This enables us to compare the use of second-hand reports to relying exclusively on first-hand observation.

2. Robustness. First, against false accusations (false positives). We consider a false positive to be the classification of a normal node as misbehaving by one normal node. Figure 3 shows them. Second, robustness against false praise (false negatives), i.e. a misbehaving node has been classified as normal despite the information available, hence there was a misclassification in the steady state of the protocol.

3. Overhead in terms of control messages, computation, and storage: The overhead directly caused by our reputation system applied to CONFIDANT is measured by the number of first-hand publications per node broadcast with a TTL of 1. It depends only on the chosen timer between publications, in this scenario the timer is set to 10s. These publications do not get forwarded. Storage overhead are the three ratings, $R_{i,j}$, $T_{i,j}$, and $F_{i,j}$, that each node i stores about each node j that it cares about. The ratings consist of two parameters each.

4.1 Liar strategies

Untrustworthy nodes can have different strategies to publish their falsified first-hand information in an attempt to influence reputation ratings, e.g. when they want to discredit normal nodes or raise the reputation of misbehaving nodes. The basic strategies consist of changing the parameter α, denoting misbehavior instances, or β, denoting normal behavior, or both. These can then also be mixed or applied only occasionally. If for example both parameters are changed by swapping them,

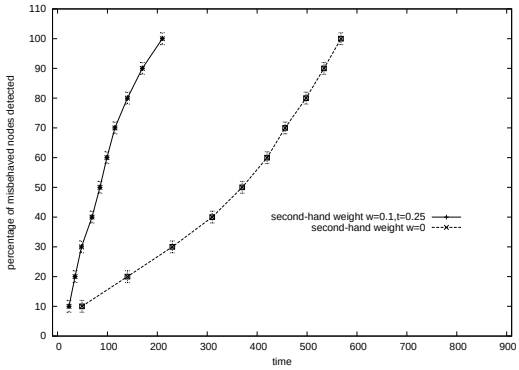


Figure 2: Mean Detection Time of All Misbehaving Nodes.

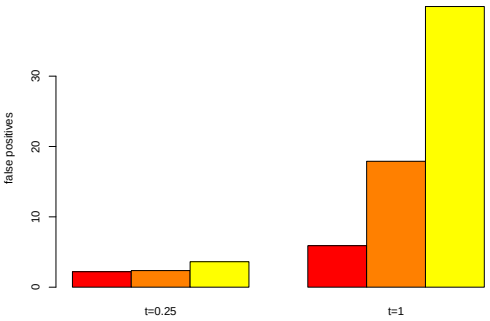


Figure 3: False Positives with Increased Untrustworthy Population, 10%, 50%, and 90%.

they will not pass the deviation test explained in Equation 10. Consider the strategy to worsen the published information about a node, the case of artificially improving it is analogous. If the worsening is considerable, it will not pass the deviation test. A more sophisticated alternative is a stealthy approach where the published information about another node is only worsened a little. Although nodes do not know the content of the reputation ratings held by others, they could try to make an inference from the first-hand information nodes publish to make an estimation. They could then try to lie only so much as to just pass the deviation test. Even when this is successful, the impact is very small as it, having passed the deviation test, only differs slightly from the reputation rating a node already has. The impact is further reduced by fading and by the limited frequency by which nodes consider second-hand information by another node.

4.2 Intoxication

If nodes use the trust option to allow incompatible second-hand information to be used in order to speed up detection, nodes could try to gain trust from others by telling the truth over a sustained period of time and only then start lying. To exacer-

bate that problem, nodes could also just reflect the second-hand information they receive from others and publish it as their first-hand information without having to have actual first-hand information themselves. We call this intoxication. This effect is mitigated by two properties of our approach. First, fading discounts trust gained in the past and recent deviations reduce trust more strongly. Second, in telling the truth or publishing whichever information passes the deviation test, they actually reinforce the reputation ratings other nodes have, making it harder to have their then deviating information be accepted.

5 Conclusions

In this paper, we proposed a robust reputation system for misbehavior detection in mobile ad-hoc networks. Our solution is based on a modified Bayesian estimation approach which we designed. In our approach, everyone maintains a reputation rating and a trust rating about everyone else who is of interest. The approach is fully distributed and no agreement is necessary. However, to speed up the detection of misbehaving nodes, it is advantageous to, cautiously, make use also of reputation records from others in addition to first-hand observations. These records are only considered in the case when they come from a source that has consistently been trustworthy or when they pass the deviation test which evaluates compatibility with one's own reputation ratings. Even after passing the test, they only slightly modify the reputation rating of a node. The results of the deviation test are additionally used to update the trust rating. We allow for redemption and prevent capitalizing excessively on past behavior by two mechanisms, namely re-evaluation and fading. It has been argued that traditional statistical approaches so far do not assume malicious behavior [8]. Our reputation system uses Bayesian estimation to specifically address lying nodes. We showed that our method is coping well with false second-hand reports, as it keeps the number of false positives and false negatives low. Our simulation also showed that the detection of misbehaving nodes accelerates significantly with the use of selected second-hand information. As a next step, we will be evaluating our reputation system as applied to a peer-to-peer network.

References

[1] Karl Aberer and Zoran Despotovic. Managing trust in a peer-2-peer information system. In *Proceedings of the Ninth International Conference on Information and Knowledge Management (CIKM 2001)*, 2001.

[2] James O. Berger. *Statistical Decision Theory and Bayesian Analysis*. Springer, second edition edition, 1985.

[3] Sonja Buchegger and Jean-Yves Le Boudec. Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes — Fairness In Dynamic Ad-hoc NeTworks. In *Proceedings of IEEE/ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Lausanne, CH, June 2002. IEEE.

[4] Sonja Buchegger and Jean-Yves Le Boudec. The effect of rumor spreading in reputation systems in mobile ad-hoc networks. Wiopt'03, Sofia-Antipolis, March 2003.

[5] Marco Carbone, Mogens Nielsen, and Vladimiro Sassone. A formal model for trust in dynamic networks. BRICS Report RS-03-4, 2003.

[6] Anthony Davison. *Statistical Models*. Cambridge University Press, Cambridge Series in Statistical and Probabilistic Mathematics, ISBN: 0521773393, October 2003.

[7] Chrysanthos Dellarocas. Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of the ACM Conference on Electronic Commerce*, pages 150–157, 2000.

[8] Roger Dingledine, Nick Mathewson, and Paul Syverson. Reputation in p2p anonymity systems. Workshop on Economics of Peer-to-Peer Systems, Berkeley, CA, June 2003.

[9] John R. Douceur. The sybil attack. In Proc. of the IPTPS02 Workshop, Cambridge, MA (USA), March 2002.

[10] Alberto Fernandes, Evangelos Kotsovinos, Sven string, and Boris Dragovic. Incentives for honest participation in distributed trust management. In *Proceedings of iTrust 2004, Oxford, UK*, March 2004.

[11] Eric Friedman and Paul Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 2001.

[12] Trusted Computing Group. Tcg main specification version 1.1b. https://www.trustedcomputinggroup.org/, November 2003.

[13] Audun Josang and Roslan Ismail. The beta reputation system. In *Proceedings of the 15th Bled Electronic Commerce Conference*, Bled, Slovenia, June 2002.

[14] R. Jurca and B. Faltings. An incentive compatible reputation mechanism. In *Proceedings of the IEEE Conference on E-Commerce, Newport Beach, CA, USA*, June 24-27, 2003.

[15] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the Twelfth International World Wide Web Conference, May, 2003*, 2003.

[16] Peter Kollock. The production of trust in online markets. *Advances in Group Processes, edited by E. J. Lawler, M. Macy, S. Thyne, and H. A. Walker*, 16, 1999.

[17] A. Halberstadt L. Mui, M. Mohtashemi. A computational model of trust and reputation. In *Proceedings of the 35th Hawaii International Conference on System Science (HICSS)*, January 7-10, 2002.

[18] G. Montenegro and C. Castelluccia. Statistically unique and cryptographically verifiable(sucv) identifiers and addresses. NDSS'02, February 2002., 2002.

[19] Tim Moreton and Andrew Twigg. Enforcing collaboration in peer-to-peer routing services. In Proceedings of the First International Conference on Trust Management, Heraklion, Crete, May 2003.

[20] Paul Resnick and Richard Zeckhauser. Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system. Working Paper for the NBER workshop on empirical studies of electronic commerce, 2001.

[21] Paul Resnick, Richard Zeckhauser, Eric Friedman, and Ko Kuwabara. Reputation systems. *Communications of the ACM*, 43(12):45–48, 2000.

[22] G. Zacharia, A. Moukas, and P. Maes. Collaborative reputation mechanisms in electronic marketplaces. In *Proceedings of the 32nd Annual Hawaii International Conference on Systems Sciences. HICSS-32.*, 1999.

[23] Xiang Zeng, Rajive Bagrodia, and Mario Gerla. GloMoSim: A library for parallel simulation of large-scale wireless networks. Proceedings of the 12th Workshop on Parallel and Distributed Simulations–PADS '98, May 26-29, in Banff, Alberta, Canada, 1998.