Title

The design of a reliable reputation system

Journal

Electronic Commerce Research, 10(3)

ISSN

1572-9362

Authors

Swamynathan, Gayatri
Almeroth, Kevin C.
Zhao, Ben Y.

Publication Date

2010-12-01

DOI

10.1007/s10660-010-9064-y

Peer reviewed

Electron Commer Res (2010) 10: 239–270
DOI 10.1007/s10660-010-9064-y

The design of a reliable reputation system

Gayatri Swamynathan · Kevin C. Almeroth · Ben Y. Zhao

Published online: 31 August 2010

Abstract Next generation Web 2.0 communities and distributed P2P systems rely on the cooperation of diverse user populations spread across numerous administrative and security domains. Zero accountability via anonymous online identities and divergent interests result in *selfish* behavior that can disrupt or manipulate networks for personal gain. While "reputation systems" are recognized as a promising means to establish social control for such communities, developing *reliable* reputation systems remains a challenge. Several unaddressed threats still limit the effectiveness of reputation systems. Furthermore, most existing work on reputations has focused on accurate reputations for stable systems, but not examined the implications of integrating user reputations into scalable distributed infrastructures. The primary goal of this paper is to investigate and address the critical open challenges that limit the effectiveness of reputations. First, we identify a thorough taxonomy on reputation management, and use it as our framework to classify adversarial threats that compromise reliable operation of reputation systems. Second, we survey existing research to address these threats. Finally, we present our solutions to address the two leading reasons for erroneous and misleading values produced by reputation systems today, *i.e.*, user collusion and short-lived online identities. We believe that this paper not only serves as an introduction to reputation systems design, but will also help researchers deploy reliable reputation solutions that contribute towards improving the performance of large distributed applications.

Keywords Peer-to-peer systems · Reputation · Security · Trust

G. Swamynathan · K.C. Almeroth (✉) · B.Y. Zhao
Department of Computer Science, University of California, Santa Barbara, CA 93106-5110, USA
e-mail: almeroth@cs.ucsb.edu

G. Swamynathan
e-mail: gayatri@cs.ucsb.edu

B.Y. Zhao
e-mail: ravenben@cs.ucsb.edu

1 Introduction

The burst in Internet connectivity around the globe in the last decade has resulted in the rapid increase in the popularity of online communities. Internet marketplaces like eBay.com witness trading of millions of unique items each day between diverse communities of individuals. Popular P2P protocols and Web 2.0 applications such as BitTorrent, Facebook, and YouTube also attract users worldwide by offering new and novel user-generated content and services. Their popularity stems from the ability to exchange information, digital content, and goods with a wide community of users not reachable through traditional means.

While these next-generation Internet communities offer a variety of opportunities, there is also risk involved for their members. These applications rely primarily on cooperative user behavior for their correct operation, a challenging task given that users are distributed over many distinct networks and administrative domains. These users are also autonomous and self-interested, behaving only in their best interests. Moreover, the availability of cheap and anonymous online identities frees them from the consequences of their actions. This open and anonymous nature that makes interacting in online communities so popular also makes them vulnerable to attacks from malicious and self-interested members.

As a popular P2P network, for example, Gnutella is susceptible to a variety of attacks [21]. One common attack is "whitewashing," where a free-riding peer repeatedly joins the network under a new identity in order to avoid the penalties imposed on free-riders. A more serious attack is when dishonest peers distribute viruses and Trojan horses hidden as files. The *VBS.Gnutella* worm, for example, stores Trojan-horse executable files on network peers [55]. Another Gnutella worm called *Mandragore* registers itself as an active peer in the network, and provides a renamed copy of itself for download in response to intercepted queries [11]. Finally, dishonest peers often pass corrupted or blank files as legitimate content.

In order to reduce such transaction risks and improve performance, applications must manage trust relationships between users, motivating cooperation and honest participation within their networks. Introducing trust to large-scale distributed applications is a difficult challenge, but one well-suited for reputation systems. A reputation system collects, aggregates, and disseminates feedback about a user's behavior, or *reputation*, based on the user's past interactions with others. Like real-world markets where personal or corporate reputations play a fundamental role in pricing goods and initiating transactions, digital reputations present a powerful mechanism to establish trust between strangers on the Internet and facilitate transactions between them.

A large amount of research confirms the fact that online reputation systems are an effective means of social management; they discourage maliciousness and motivate trustworthiness and cooperation among users [1, 6, 28, 40, 45, 57]. Most existing work on reputations, however, has focused on accurate reputations for stable systems, but not examined the implications of integrating user reputations into scalable distributed infrastructures. For instance, existing reputation systems provide misleading results for unstable and "short-lived" user identities, a commonly observed phenomenon in dynamic distributed systems. Since reputations assess a user's trustworthiness using historical feedback of its past interactions, longer user lifetimes lead

to more interactions, and a more accurate reputation. But users in "high-churn" systems are often short-lived as they periodically exit the application or leave due to failures. Furthermore, malicious users penalized by reputation systems for poor performance have the ability to rejoin the network with newly acquired identities and a clean history. Such users accrue inaccurate reputations computed from only a small number of past interactions.

On the other hand, "long-term" reputations, aggregated from a larger number of past transactions, are challenged with another serious threat—vulnerability to user collusion. Reputation systems generally assume that each online identity represents a single user. However, recent work has shown that given the relative low cost of online identities, users often generate multiple "Sybil" identities to gain benefits beyond the fair allocation for a single identity [59]. The Sybil attack, as this is popularly known, also allows these multiple identities to "collaborate" or collude for the good of the user. For example, users can collude to artificially boost the reputation values of one or more friends [34], or falsely accuse well-behaved users of misbehavior. Detecting such collusion attacks is yet an unsolved problem that severely limits the impact of existing reputation systems.

The primary objective of this paper is to investigate and address the critical open challenges that limit the effectiveness of reputations and prevent their integration into large-scale distributed applications today. Integrating reliable reputation solutions will contribute tremendously towards increasing user cooperation, thereby improving the performance of these applications. Towards this end, this paper first identifies a thorough taxonomy on *reputation management*, namely, the tasks of collection, aggregation, storage, and communication of reputation data. Our goal is to employ this taxonomy as a framework to facilitate two specific contributions: identify challenges that compromise reliable operation in each category, and survey prominent strategies to overcome these challenges. Furthermore, we present our contributions towards addressing the two critical reputation reliability challenges that remain largely unsolved today—collusion and churn attacks.

First, we counter user collusion by augmenting traditional reputation systems with a "reliability metric" [54]. Our approach helps users make more accurate decisions based on trust by quantifying the risk that a given reputation value has been affected by collusion or collusion-like behavior. As a basis for our metric, we leverage a pair of well-studied mechanisms used in economic studies, the Lorenz curve [35] and the Gini coefficient [10]. Applied to reputations, they characterize how far a user's per-partner distribution of transactions deviates from the ideal. Using our metric, a user can easily distinguish between reputations generated by truthful transactions and those that might be strongly influenced by user collusion.

A user deemed "unreliable" by our metric could either have transacted exclusively with a small number of partners, or performed very few transactions, possibly due to having just joined the network. For such users, we describe the use of *proactive reputations*, a technique to obtain accurate, firsthand estimates of a user's reliability [51]. Our experimental evaluations demonstrate how these firsthand observations are resistant to both churn attacks and user collusion.

The remainder of this paper is organized as follows. We begin by presenting an overview of reputation systems and our taxonomy on reputation management in

Sect. 2. We also present a summary of six representative reputation systems. Section 2 also describes each aspect of reputation management, identifying challenges that compromise reliable operation and surveying research efforts to overcome these challenges. In Sects. 3 and 4, we discuss the critical challenges of collusion and churn, and describe our solutions towards addressing them. First, we describe our reliability metric for reputation systems that protects users from collusion attacks (Sect. 3). We also describe the use proactive reputations to compensate for unreliable reputations. Second, we evaluate both mechanisms via detailed simulations of peer collusion and churn models based on measurements of deployed P2P systems, and show how our enhancements greatly improve the accuracy of traditional reputation systems under collusion and churn (Sect. 4). Finally, we present related work in Sect. 5 and conclude the paper in Sect. 6.

2 Taxonomy on reputation management

A reputation is a statistical estimate of a user's trustworthiness computed from feedback given by previous transaction partners to the user. As depicted in Fig. 1, a service requester, R, uses the reputation profile of a provider, P, to determine whether to transact with P. Following the transaction, R provides its assessment (or rating) of P to the reputation system. Examples of rating schemes include binary ratings (0 indicating bad, 1 indicating good) or subjective ratings like (Very Good, Good, OK, Bad, Very Bad).

The predictive power of reputation assumes that past behavior of a user is indicative of her future behavior [31]. A user enjoying a highly valued reputation would imply that the user has conducted good transactions in the past, and can be trusted

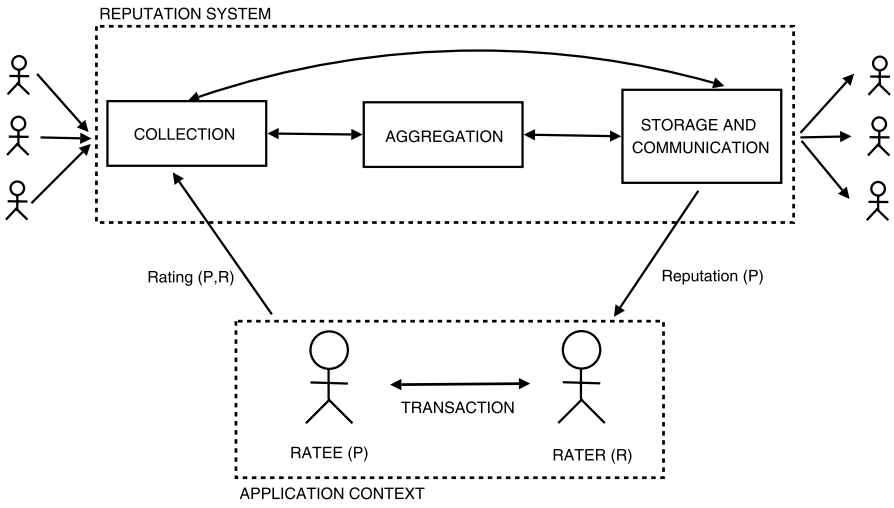

Fig. 1 A reputation system collects, updates, and disseminates each peer's behavior in the network in the form of its reputation profile

to perform honestly in future transactions as well. eBay.com, for example, one of the largest marketplaces on the Internet today, uses a feedback-based reputation system where buyers and sellers rate each other following each transaction. The eBay reputation system (the Feedback Forum) computes a user's reputation as the sum of its lifetime (positive, neutral, negative) ratings. Sellers with excellent reputations can claim higher prices for their products while poor reputation holders attract fewer buyers [43].

We identify a taxonomy of four reputation management processes that need to be effectively designed in order to build a reliable reputation system, namely, the collection, aggregation, storage, and communication of reputation data. We employ this taxonomy as a framework to organize recent research in reputations as well as examine the various strategies proposed by the research community to ensure reliable operation of reputation systems. In the *collection* process, we describe the generation of ratings by users undertaking transactions. We discuss the sources and types of ratings, and the various ratings schemes that can be employed by a reputation system. In the *aggregation* process, we describe how individual user ratings can be aggregated to form a reputation profile. This reputation profile needs to efficiently stored in order to minimize communication and processing overheads. In the *storage* process, we discuss the various forms in which reputation can be stored, the choice of data store, and storage methodology. The *communication* process discusses the various reputation exchange protocols that can be employed.

Each of these components needs safeguarding against a variety of adversarial threats. For example, reliability in terms of reputation accuracy is a critical requirement from the aggregation component. Reliability guarantees like privacy and availability are critical from the storage component managing reputation data. Likewise, the communication infrastructure should be resistant to packet sniffers who attempt to maliciously tamper with information and present false information to reputation seekers. This paper, therefore, studies the extent to which existing research efforts counter such threats.

We now present an overview of the prominent P2P reputation systems analyzed in this paper. Next, we describe each aspect of reputation management in detail and survey related work. While the area of P2P networks has experienced a predominant amount of reputations research, Sect. 5.1 discusses other applications of reputation systems, including eCommerce marketplaces, Web 2.0 applications, and wireless networks.

Summary of P2P reputation systems We analyze six major reputation systems, namely, PGrid [1], XRep [11], EigenTrust [28], CONFIDANT [6], TrustGuard [49], and Credence [56]. We choose to compare these systems not only because they have significantly shaped the area of reputations research over the last five years, but also because of the diversity in their reputation mechanisms that we wish to analyze. We discuss the strengths and weaknesses of each of these reputations systems with respect to our reputation management taxonomy. In addition to our six representative systems, we discuss other P2P reputation systems that present interesting solutions for specific process contexts in the text when appropriate.

Aberer and Despotovic were one of the first to propose a reputation system for decentralized networks [1]. Their approach, PGrid, assumes that network peers are

honest in most cases. The reputation system expresses distrust in the form of *complaints* and a simple summarization of complaints received and filed is performed. The authors use probabilistic analysis to compute an average trust measure (based on number of complaints) for network peers and determine dishonest peers as ones which exceed the average trust value. Decentralized data management is achieved using a PGrid.

The XRep protocol, proposed by Damiani et al., is a reputation sharing protocol proposed for Gnutella, where each peer keeps track and shares with others the reputation of their peers [11]. A combination of peer and object reputations are used to minimize the potential risk involved with the download and use of a resource. A distributed polling algorithm is employed to manage the reputations.

Kamvar et al. propose EigenTrust, a reputation system to help P2P file sharing networks combat the spread of inauthentic files [28]. Each peer is associated with a *global* trust value that reflects the experiences of all the peers in the network with that peer. Eigentrust uses a distributed algorithm where these global trust values are iteratively aggregated along transitive trust chains of *local* trust values weighted by the reputation rating of the raters.

Buchegger et al. propose CONFIDANT, where each node monitors its neighbors' behavior and maintains a reputation for each neighbor [5, 6]. The authors attack the problem of false ratings by using a Bayesian approach. They distinguish between reputation, how well a node behaves in routing, and trust, how well it behaves in the reputation system. A node distributes only firsthand information to other nodes, and only accepts other firsthand information if those opinions are similar to its own opinion.

Srivatsa et al. propose TrustGuard to counter three vulnerabilities identified by them as detrimental to decentralized reputation management, namely, oscillatory peer behavior, fake transactions, and unfair rating attacks [49]. TrustGuard employs a *personalized similarity measure* (previously proposed by the same authors in PeerTrust [57]) in order to more heavily weigh opinions of peers who have provided similar ratings for a common set of past partners, thereby, addressing the problem of dishonest feedback. The PeerTrust trust model also considers the transaction context (on the basis of transaction size, category or time stamp) and incentives to provide feedback.

Walsh et al. propose Credence with the goal of thwarting file (or object) pollution in P2P file-sharing networks [56]. Similar to XRep, the authors generate object reputations and deploy their mechanism for files in the Limewire client for the Gnutella P2P network [21]. Credence employs a web-of-trust to account for the lack of direct observations. This is because it is impossible for a single object to be widespread enough to have a sufficient number of raters for it. Like PeerTrust's personalized similarity metric, Credence employs a *correlation coefficient* to compare voting histories between peer pairs.

Table 1 summarizes these reputation systems. We now begin a detailed discussion of each aspect of reputation management.

Table 1 Summary of P2P reputation systems

Reputation system	Collection	Aggregation	Storage and communication
PGrid [1]	Only complaints are reported	Simple summarization of complaints	Decentralized P-Grid
XRep [11]	Binary votes (for peers and resources)	Distributed polling algorithm; no formal trust metric	Local repositories maintained by users; Vote polling via broadcast
EigenTrust [28]	$(-1, 0, +1)$ ratings	Transitive trust chains of *local* trust values weighted by the rating of the raters aggregated to form global trust values	DHT-based structured approach
CONFIDANT [6]	ALARM messages warn users	Decoupled service and feedback trust; Bayesian approach of aggregating "similar" ratings	Trust table maintained by each user; Control messages exchange reputation data
TrustGuard [49]	Binary ratings	Decoupled service and feedback trust; Personalized "similarity" metric with cached and dynamic computations	P-Grid-based approach
Credence [56]	Only resources voted $(-1, +1)$	Correlation "coefficient" compares voting histories of user pairs	Local vote databases; Vote gathering protocol

2.1 Reputation collection

Peer-to-peer networks are defined by interactions between peers. An interaction between two peers could include a file transfer, remote storage, CPU usage, or similar transactions. The service requesting peer uses reputation information to decide among multiple providers. At the end of the transaction, the requester furnishes the system with its evaluation of the service provider in the form of a reputation rating. Examples of quantitative rating schemes include a binary rating scheme (0 indicating bad, 1 indicating good) [57], or a $(-1, 1)$ scheme where -1 indicates a bad transaction, and 1 indicates a good transaction [28]. Mobile ad hoc nodes use ALARM-type messages to warn others of malicious nodes [6].

Reputation ratings are normally associated with peers. In Credence, a reputation rating is associated with a resource instead such as a file in a file-sharing network [56]. Similarly, XRep employs a combination of peer-based reputations and resource-based reputations [11]. Damiani et al. present an elaborate discussion on shortcomings of using a pure peer-based reputation scheme or resource-based reputation scheme. Peer-based reputations suffer from the *cold-start* problem for peers; newcomers offering a limited number of resources need to struggle initially to build their reputations. By distributing well-reputed resources, newcomers can more quickly participate actively in the network. This also improves load balancing by directing peers to multiple copies of a well-reputed resource. Also, resources, and their reputations, are generally more persistent than peer identities in the system.

There are several disadvantages of using a pure resource-based reputation scheme. New files will need to build a reputation only after a file transfer (cold-start problem for files). However, with a peer-based reputation scheme, new resources offered by a well-established peer will be regarded as reliable. A major problem of using just a

resource-based reputation scheme is in the identification of malicious peers. There is no effective way of linking a bad resource to the peer that provided it. In a peer-based scheme, peers can be easily blacklisted. However, since reputations are associated with pseudonyms, peers have the ability to change identities and discard their poor reputations.

Combining peer-based and resource-based reputations helps take care of the shortcomings of any one approach. However, storage overheads are substantially higher when incorporating resource-based reputations as the number of resources in any system is significantly more than the number of peers. Also, a single resource cannot be widespread enough to have a sufficient number of raters for it. The choice of using peer-based reputations or resource-based reputations could also depend on the churn rate for each entity. If the turnover of peers in the network is high due to free and easily generated identities, then resource-based reputations may turn out to be a better option. On the other hand, if the turnover of resources in the network is high, then it may be better to leverage the more stable nature of peers.

2.2 Reputation aggregation

Reputation systems aggregate peer feedback using two approaches. One approach is to use only firsthand information to evaluate peers. That is, each peer does not consider any other peer's feedback or observations. Ratings are aggregated locally with no global information. The second approach is to use global information. Feedback from all the peers that have interacted with a peer are aggregated in a meaningful way to compute the peer's reputation. The trade-off is between the efficiency of using available reputation information and vulnerability to false ratings. While global reputation is efficient and helps quickly detect misbehavior in the system, it is vulnerable to false ratings. On the other hand, reputation ratings directly derived from firsthand experience are highly reliable, but do not help blacklist malicious peers for others. Also, firsthand information only proves effective if a peer locates honest service providers with which to repeatedly transact [36].

Since global reputations provide significantly more information than firsthand reputations, reputation systems predominantly employ them. We enlist some types of ratings misbehavior commonly observed due to global reputation aggregation.

- *Dishonest raters.* An *honest* peer is one that is honest in its ratings of other peers. A *dishonest* peer, on the other hand, tries to subvert a system by falsely rating a bad transaction as good, and vice versa. Such unfair ratings presented due to jealousy, competition, or other malicious reasons adversely affects the quality of reputation scores generated by a reputation system.
- *Dynamic personalities.* Some peers exhibit a dynamic personality, switching between honest and dishonest behavior. Behavior changes can be based on the type or value of the transaction or the party involved at the other end. Reputation *milkers*, or *oscillating* peers, attack a reputation system by first building a good reputation and then taking advantage of it to do harm.
- *Collusion.* Collusion occurs when two of more peers collectively boost one another's reputations or conspire against one or more peers in the network. Dellarocas

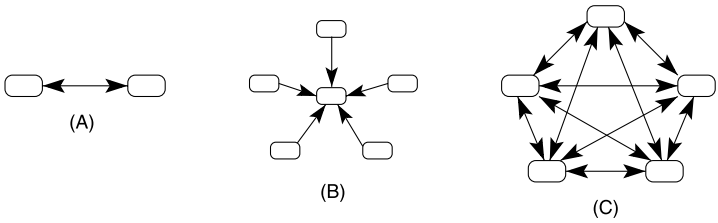

Fig. 2 Three different collusion models. (**A**) Two-user; (**B**) Sybil-based; (**C**) Group

identifies four types of collusion misbehavior [12]. In *ballot stuffing* form of collusion, a colluding group inflates each other's reputations which then allows them to use the good reputation to attack other system peers. Another form of collusion is *bad-mouthing* where a malicious collective conspires against one or more peers in the network by assigning unfairly low ratings to the target peers, thereby hurting their reputation. Finally, positive (and negative) *discrimination* arises when peers provide good (and poor) service to a few targeted peers.

– *Sybil-based collusion.* The Sybil attack occurs in the absence of a centrally trusted party, when a peer with sufficient resources can establish a potentially unbounded number of distinct online identities (or *Sybils*) [16, 59, 60]. Prior work has shown that users can use these identities to collude and artificially inflate their own reputations in order to monopolize service, lure users into scams, or otherwise gain performance benefits from the system [4]. Figure 2 depicts three common types of collusion, including, Sybil-based collusion, collusion between two users, and group-based collusion involving more than two users.
– *Churn attacks.* While reputations have been deployed in online marketplaces such as eBay.com, they are not necessarily a natural fit for the dynamic nature of P2P overlay networks. Since reputations assess a peer's trustworthiness using historical feedback of its past interactions, longer peer lifetimes lead to more interactions, and a more accurate reputation. Distributed communities like P2P file sharing networks, however, experience significant churn (or peer turnover) which means a high percentage of peers will have relatively "short-term" reputations accrued from a small number of past interactions. For instance, malicious peers penalized by reputation systems for poor performance have the ability to rejoin the network with newly acquired identities and a clean history. Such churn attacks result in erroneous or misleading reputations for malicious peers.

While an innumerable variety of attacks can be devised by malicious peers, our above set comprise attack strategies most commonly observed by reputation systems. We now discuss related work to counter these attack strategies. Our discussion is largely qualitative due to the lack of a uniform experimental infrastructure to compare the various systems. Kerr and Cohen propose a testbed formulation designed to support systematic experimentation and evaluation of reputation systems [29]. However, the model is based on vulnerabilities investigated by the authors in online marketplaces, and its application to P2P systems and other applications is not known.

Reputation estimation methods based on a simple summation are particularly vulnerable to ratings attacks. For example, the overall reputation of a participant in

eBay.com is computed as the sum of $(+1, 0, -1)$ transaction ratings acquired by the participant over its lifetime [43]. Such simple schemes result in the "increased trust by increased volume" vulnerability, *i.e.*, a peer could increase its trust value by increasing its transaction volume, thereby hiding the fact that it frequently misbehaves at a certain rate. For example, a peer could undertake a thousand good transactions of low value (say, worth $1) and use the accumulated good reputation towards one dishonest transaction of high value (say, worth $1000). Additionally, all the ratings are given an equal weight which encourages Sybil attacks and collusion.

Clearly, simple summarization schemes are ineffective given the sophisticated types of attacks presented by dishonest individuals and groups. We now discuss more interesting solutions proposed by the research community to counter the prominent attack strategies identified by us.

Dishonest raters and dynamic personalities The impact of false ratings is mitigated by incorporating credibility of the feedback source while processing a reputation rating. If Peer *A* trusts Peer *B* and Peer *B* trusts Peer *C*, then Peer *A* trusts Peer *C*. Josang et al. describe requirements for the validity of such transitivity by expressing semantic constraints under which trust may be transitive [26]. Furthermore, subjectivity is of concern to reputation systems built on transitive *web-of-trust* models. Peer *A* may regard 0.8 in an interval of $[0, 1]$ as a very high value of trust while Peer *B* may perceive this value as only average. Hasan et al. discuss solutions to eliminate such subjectivity from web-of-trust systems [23].

Several reputation systems employ web-of-trust chains to establish and propagate trust among peers [33]. In general, a longer chain implies a greater risk of encountering a malicious "link." Some schemes weigh ratings of a transitive chain by the reputation of the least reputed peer in the chain [17] or proportionally weigh down ratings as the length of the chain increases [7]. EigenTrust, a mechanism similar to PageRank [41], uses a distributed algorithm where global trust values are an aggregation of local trust values weighed by the reputation rating of the raters [28]. The algorithm, however, requires strong coordination between peers and while effective at countering false raters, the approach is complex and a more general solution is needed.

EigenTrust and PGrid are "coupled" trust approaches—they correlate service trust to imply feedback trust. That is, peers reputed to provide trustworthy service likely provide trustworthy feedback. The feedback from peers with higher credibility, consequently, weighs more in the calculation of a reputation score than those with lower credibility [1]. While useful as a simple defense, such a mechanism can easily fail or be manipulated. For example, colluding nodes can offer honest service for the express purpose of boosting their reputations so they can badmouth other peers.

An alternative to coupled (or correlated) trust is to build a separate trust metric to evaluate the credibility of feedback [6, 53, 57]. Feedback from peers with higher feedback trust ratings will have more impact on a reputation score than those with lower feedback ratings. But this technique requires additional overhead and computational complexity. For instance, TrustGuard uses the root mean square or standard deviation to determine dissimilarity in the feedback ratings between any two peers, thereby, determining the likelihood of credibility of each other's opinions [49]. In a

large P2P system, however, finding a statistically significant set of such past partners is a challenge. Peers are likely to make choices among a set of candidates for which there is no information.

CONFIDANT, another decoupled trust mechanism, distinguishes between reputation, *i.e.*, how well a node behaves in routing, and trust, *i.e.*, how well it behaves in the reputation system. A node distributes only first-hand information to other nodes, and only accepts other first-hand information if those opinions are similar to its own opinion. Compared to CONFIDANT, where a node's referral is interpreted subjectively per node, Swamynathan et al. produce a system-wide referrer rating per node making it more scalable [53]. The authors also demonstrate how the lack of trust data can impact the effectiveness and scalability of TrustGuard in its computation of trust values.

Both coupled and decoupled trust models need to take into account the distribution of peer feedback in the computation of global reputation values. Zhou et al. propose PowerTrust, a reputation system that, by taking into account the power-law characteristics of feedback distributions, observes improvements in reputation accuracy and speed of reputation aggregation [65].

Finally, to deal with dynamic personalities, reputation systems commonly employ reputation *fading* or decay. By weighing feedback from recent transactions more than old transactions, a peer is forced to maintain its honest behavior. This idea also helps previously malicious participants shed their poor reputation, and with time, re-build a good reputation.

Peer collusion Dellarocas identifies ballot-stuffing and bad-mouthing as two colluding scenarios in which peers can intentionally try to "rig the system" with biased reputations, and demonstrates that while controlled anonymity can be used to avoid bad-mouthing, cluster filtering approaches help reduce ballot-stuffing [12].

PageRank, one of the most popular reputation systems for ranking and web search today, can be easily manipulated by collusion and Sybil strategies [41]. The EigenTrust algorithm applies PageRank for P2P networks, but attempts to address the problem of malicious collectives by assuming pre-trusted peers in the network which may not be a practical consideration. Lian et al. demonstrate the vulnerability of Eigentrust to collusion by applying the algorithm to the Maze P2P system data set [34]. Zhang et al. improve eigenvector-based reputations by capturing the amount of PageRank inflation obtained by collusions [62]. They observe that colluding nodes cheat the algorithm by stalling the PageRank random walk in a small web graph, and hence, these nodes are sensitive to the reset probability of the random walk.

Sybil attacks All distributed reputation systems are vulnerable to Sybil attacks; peers can generate a large number of identities and maliciously increase the reputation of one or more master identities by giving false recommendations to it. Dewan et al. suggest such liar farms can be countered if all the identities of a peer can be mapped back to it [13]. They propose an IP-based mechanism that defines a security zone and averages all the recommendations received by identities whose IP lie in the same security zone. Similarly, the Maze system counters the Sybil attack by employing a combination of IP address and hard-drive serial IDs to track machines [58]. Other schemes, like OpenPrivacy, use identity plugins [7].

Cheng et al. show that symmetric reputation functions cannot be resistant to Sybil attacks as web-of-trust subgraphs can be duplicated by malicious peers to raise their reputations arbitrarily. Sybilproof reputation mechanisms, consequently, need to design asymmetric reputation functions [9]. As with collusion, eigenvector algorithms are extremely vulnerable to the Sybil attack as peers can increase their reputation values by creating complete subgraphs of Sybil identities. Finally, the SybilGuard protocol is based on the idea that malicious users can create several identities but fewer trust relationships [59, 60]. The disproportionately-small "cut" in the social network graph between the Sybil nodes and the honest nodes is exploited to bound the impact of multiple identities.

Malicious colluders and Sybils present a significant challenge to reputation systems design. Our previous work proposed a "reliability metric" to detect and penalize collusion-like behavior, and encourage peers to interact with diverse groups of users across the network. This metric leverages two mechanisms used in economic studies, the Lorenz curve [35] and the Gini coefficient [10], which characterize how far a user's per-partner distribution of transactions deviates from the ideal uniform distribution [54]. We discuss our solution in greater detail in Sects. 3 and 4.

Churn attacks High rates of peer turnover, or *churn*, means a significant percentage of peers will have relatively "short-term" reputations accrued from a small number of past interactions. For applications that rely on peers for data storage, message forwarding, or distributed computation, choosing a peer based on short-term reputations is highly undesirable. This fundamental reliability concern greatly limits the effectiveness of existing reputations mechanisms and their potential impact in network protocols and distributed applications today.

The availability of cheap identities results commonly in the whitewashing attack presented by free-riding (or selfish) peers. A free-riding peer conserves bandwidth and CPU by not contributing any resources to the system. Various incentive schemes have been proposed to encourage cooperation and participation in the network [17, 18]. One proven way for a system to deal with high churn is to distrust all newcomers in the system [19]. However, with such a mechanism, legitimate newcomers are treated poorly initially, at least until they build a positive reputation. Feldman et al. suggest a "stranger adaptive" strategy to counter whitewashing in a network [17]. Using recent transactions with strangers, a peer estimates the probability of being cheated by the next stranger, and decides whether to trust the next stranger using that probability. Swamynathan et al. explore proactive firsthand reputations as a solution to generate quick and reliable reputations for short-lived network peers [51]. We discuss this solution in greater detail in Sects. 3 and 4.

2.3 Reputation storage

Different applications use different storage schemes that determine how data is inserted, accessed, and updated. Because any central storage approach would limit the scalability of a peer-to-peer system, reputation data needs to stored in a decentralized fashion. Decentralized storage can be achieved by having reputation data stored by the provider [13, 40], the requester [11], or an anonymous third-party [1, 28].

Chord [50], CAN [42], Tapestry [63] and PGrid [1] use a Distributed Hash Table (DHT) that deterministically maps keys into points in a logical coordinate space. Searches, as well as storage space required at each node, are on the order of $\log N$. Anonymity and redundancy mitigate peer collusion and tampering of reputation. In PGrid, security concerns can arise if a peer stores its own trust information. However, the authors find this occurrence rare and propose redundancy to ensure data integrity.

An alternative to the structured storage mechanism is to have each peer store trust values locally [11, 13]. *PRIDE* employs an *elicitation-storage* protocol that cryptographically prevents malicious modification of reputation information [13]. Peers in XRep maintain a cryptographically-secure experience repository of resources and "servents" with which they have interacted [11]. Certificates are another common way of storing trust values. To prevent tampering, certificates can be digitally signed with the private key of the certificate creators. Certificates can be stored at the creator and the target [7], or by the target alone [40]. The tradeoffs to storing reputation data at only one location are the communication and processing overheads involved in ensuring the integrity and authenticity of the data.

2.4 Reputation communication

The data exchanged, the storage mechanism employed, and the type of peer-to-peer network (structured or unstructured) are some factors that determine the type of communication protocol that can be employed. XRep, the reputation sharing protocol proposed for Gnutella, employs vote polling via broadcasting on the Gnutella network [11]. *Poll* messages are implemented on top of ordinary *Query* messages. To protect the integrity and confidentiality of poll responses, the poll request includes a public key, generated on the fly, with which poll responses need to encrypted. Similarly, Dimitriou et al. describe SuperTrust, an encryption-based framework to preserve the privacy and anonymity of transaction ratings [14]. In DHT-based storage approaches, a peer is responsible for multiple keys and also the maintenance of a routing table for other keys [1, 57]. When the peer receives a search or an update message with a data key that it is not responsible for, it forwards the request according to its routing table. Searches take $O(\log N)$, where N is the number of peers in the network. Encryption schemes can be incorporated for secure transmission of data [57]. Another issue in reputation systems is to determine between cached and dynamic computations of user reputations. Dynamic computation of data is expensive if a peer has to retrieve the trust data of several peers in the network at run time. Each peer can alternatively maintain a trust cache that stores the most recent trust values of peers with which it has interacted [57]. Data retrieved from the cache results in only *approximate* computations, but it is a cost-effective solution.

Issues like storage and communication integrity of reputation data are critical for building reliable reputation systems, but these are generic challenges encountered by most distributed infrastructures and are not a particularly novel problem for reputation systems. Reputation systems can be designed to leverage secure storage and communication protocols implemented by the underlying distributed application. We refer readers to [30, 40] for a more complete analysis of design challenges in distributed systems storage and communication.

As pointed out earlier in this section, two critical challenges still hinder reputation systems in detecting and penalizing malicious users. These are the challenges posed by user collusion and high churn in large-scale dynamic systems. The following sections now describe our solutions to address these open challenges.

3 The challenges of collusion and churn

Despite their effectiveness in traditional controlled environments, current reputation systems can be highly inaccurate in large dynamic networks, such as online communities. One of the leading factors to inaccurate reputation values is the increasing presence of user collusion behavior. First, most online communities lack strong authentication, allowing users to obtain multiple "independent" online identities. Prior work has shown that a user can use these identities to collude and artificially inflate his own reputation to monopolize service, lure users into scams, or otherwise gain performance benefits from the system [34]. A second issue deals with "short-lived" peers who have conducted few transactions. Since reputation systems rely on sampling prior transactions, few transactions mean a small sample set and a reputation that is easily influenced by malicious partners. Furthermore, malicious users penalized by reputation systems for poor performance have the ability to rejoin the network with newly acquired identities and a clean history. Such users accrue inaccurate reputations computed from only a small number of past transaction.

We address these issues of reputation accuracy using two complementary reputation mechanisms. First, we introduce a "reliability" metric that estimates the *accuracy* of a network reputation based on the observation that reputation values are most accurate when computed from numerous past transactions distributed across many distinct partners. Second, for a user that determines that its service partners do not have reliable reputations, we describe how to leverage "proactive reputations" to produce reputation values with the desired accuracy.

3.1 Understanding user collusion behavior

Before defining our collusion-resistant metric, we need to first clearly define our collusion attack model. We begin this section by quantifying the potential impact of collusion behavior on system-wide performance. We then describe our assumptions and models for colluding attackers, with models drawn from previous measurement studies.

Impact of user collusion To better understand the threat that collusion attacks pose to reputation systems, we perform an experiment using an event-driven simulator where random subsets of a network of 10,000 peers collude to improve their reputation values. We define reputations as values between 0 and 1, where 0 indicates no trust, and 1 indicates absolute trust. For each peer, we define an "intrinsic trust value" that guides the peer in its transactions. For example, a peer with an intrinsic trust value of 0.8 has a random 80% chance of behaving honestly on any given transaction. We set malicious peers with trust values less than 0.3. We then allow random peer pairs

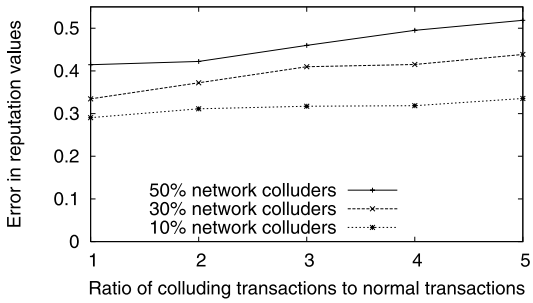

Fig. 3 Impact of user collusion on perceived reputations

to perform transactions in the system, with the subsequent feedback recorded to compute the participants' reputations. We assume a uniform distribution of transactions with an average of 15 normal transactions initiated per peer. In addition to these normal transactions, we allow a subset of 2–5 peers to perform collusion by performing transactions within the group which is always followed by mutual positive feedback. Figure 3 plots the collusion-induced absolute error values for affected peers as computed by the difference in reputation values with and without colluding transactions. Clearly, even a relatively low rate of collusion can have a dramatic impact on a peer's perceived reputation values.

Collusion model Our collusion model begins with two assumptions. First, we assume that peers cannot modify the application, and must provide verifiable proof of a transaction along with its transaction feedback. This prevents colluders from spoofing an unlimited number of transactions, and can be achieved using reasonable secure signature mechanisms. Second, we assume that while colluders cannot forge transactions, they can perform collusion transactions with resource costs lower than legitimate transactions. For example, data transfers between two application instances on the same machine generally incur much lower processing and I/O overhead compared to typical transactions between distant peers. To model the lower cost of collusion transactions, we use a *collusion cost factor* to represent the ratio of resource costs between a legitimate transaction and a colluding transaction. We use this factor to estimate the number of illegitimate transactions that can be reasonably performed by colluders in our experiments.

To accurately evaluate our metric, we require a test framework with realistic models of user collusion. For this purpose, we leverage the results of a recent measurement study on the Maze peer-to-peer file-sharing network that showed user behavior strongly indicative of multi-user collusion. Maze is a popular file-sharing system in Asia, and uses a centralized architecture that logs all transactions, crediting users for each successful file upload while consuming credits for downloads based on file size [58].

This study examined a complete log of the Maze system over a period of one month, including 32 million file transfers totaling more than 437 terabytes between 161,000 users [34]. It observed several types of highly probable collusion-like behavior, including how multiple peers performed repetitive or faulty transactions to artificially inflate the download credits of certain peers. The results support the prevalence

of three popular collusion models. We use these models to drive the test framework used in Sect. 4. We illustrated these models in Fig. 2, and describe them below:

– *Pairwise collusion.* The simplest model where two peers collude to mutually boost reputation values, *e.g.*, repeatedly download the same content from each other. This can be performed by two distinct users, or by two Sybil identities.
– *Sybil-based collusion.* A single user boosts its reputation with help from a large number of "slave peers" obtained via a Sybil attack [16]. Slaves exist only to transact with the "master peer" and improve its reputation.
– *Group-based mesh collusion.* Finally, multiple peers can form cliques where all members collaborate to mutually boost reputation values. Peers maximize their benefit by performing pairwise collusion with all other peers in the clique. While the aggregate benefit increases with clique size, clique sizes are limited by non-trivial maintenance and coordination costs.

3.2 A statistical reliability metric

To quantify the likelihood that a reputation value has been influenced by possible collusion, we propose a peer reliability metric based on the distribution of transactions among a peer's partner set. A reputation is less reliable if a significant fraction of transactions are performed with a small number of peers, and "more reliable" when all transactions are distributed evenly across many distinct partners. Intuitively, we can compute such a reliability by representing a Peer P's reputation as a Cumulative Function (CF) of its transaction history. That is, if we plot on the x-axis the cumulative percent of P's distinct partners (sorted by number of transactions undertaken with P) and on the y-axis the cumulative percent of P's transactions, then the most reliable distribution is represented by the 45 degree line.

Figure 4 plots transaction distributions of 3 peers that each conduct 100 transactions with 20 peers. A peer maximizes its reputation reliability by spreading its transactions evenly across all 20 peers in the system (shown by Distribution A). A colluder who performs 82% of its total transactions with two colluding partners obtains a much lower reliability value for the same total number of transactions (Distribution C). Finally, an average peer might obtain a partner distribution better than the colluder (Distribution B).

We investigated the effectiveness of several different measures as potential reliability metrics. Our search led us to the area of economic statistics, where statistical

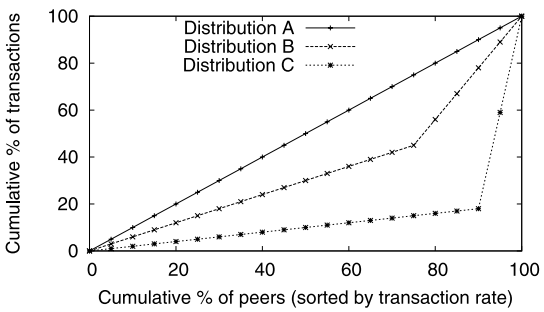

Fig. 4 The cumulative transaction percentage for reputation reliability

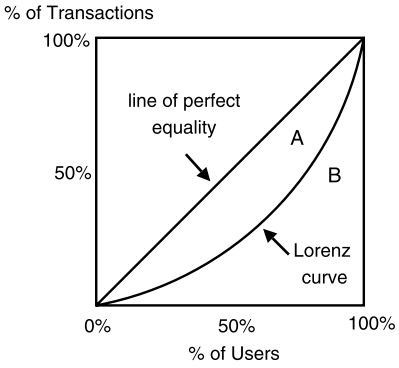

Fig. 5 A Lorenz curve representing the proportionality of a distribution

models are used to compute and compare the proportionality of such distributions. The Lorenz curve [35], in particular, is a graphical representation of the cumulative distribution function of a probability distribution. Developed by Max Lorenz in 1905, it is used in economics and ecology to describe inequality in income or size (for example, bottom $X\%$ of society has $Y\%$ of the total income). As shown in Fig. 5, the Lorenz curve of a given dataset is compared with the *perfect equality line*. In our case, this represents a perfect distribution of transactions among a peer's entire transaction partner set. The further the Lorenz curve lies below the line of equality, the more skewed is the distribution of transactions. Formally, the Lorenz curve can be expressed as:

$$Z(y) = \frac{\int_0^y x dF(x)}{\mu}, \tag{1}$$

where $F(x)$ is the cumulative distribution function of ordered individuals and μ is the average size. The total amount of inequality is summarized by the Gini coefficient [10] (G). The Gini coefficient of a given data set is the ratio between the area enclosed by the line of equality and its Lorenz curve, and the total triangular area under the line of equality. That is:

$$G = \left(\frac{A}{A+B}\right). \tag{2}$$

The Gini coefficient ranges between 0 to 1. 0 corresponds to perfect equality, *i.e.*, all partners have had the same number of transactions with the given peer. 1 corresponds to maximum inequality, *i.e.*, all transactions were undertaken with one single partner. Since higher values are favored by our metric, we compute reliability (or reputation quality) from the Gini coefficient as:

$$Q = (1 - G). \tag{3}$$

Here, Q denotes a peer reputation's reliability score. We performed a detailed experimental evaluation of this metric and report a subset of our results in Sect. 4.

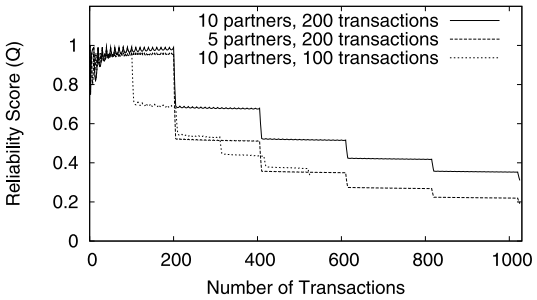

Fig. 6 The impact of collusion on the reliability of an inflated reputation

We note that colluders seeking to boost their aggregate reputation value can easily achieve a high reputation reliability (Q) at the same time, by distributing its transactions evenly between its colluding partners. This tactic fails, however, when a colluder actually seeks to make use of its reputation by cheating (and interacting) with a normal user. The more a user colludes with her friends to inflate her reputation, the more significant her drop in reliability after interacting with a non-colluder. In Fig. 6, we show how the reliability values of three colluders change as they periodically interact with honest peers. Each colluder starts by building their reputation through collusion, then goes through periodic phases of interacting with normal users followed by more collusion. We compute each colluder's reliability score, Q, after each transaction. During collusion, the colluder cycles through its partner set in a round-robin fashion to evenly distribute its transactions among them.

As Fig. 6 shows, transacting uniformly with its partner set produces perfect reliability scores for each user. However, the scores fall dramatically when colluders interact with non-colluders. Reducing the number of colluding partners or transactions per partner does not result in any improvement in reliability scores for the colluder. Once a reputation's reliability drops, it is hard to re-build it. Therefore, a user that colludes frequently with a single partner is permanently damaging her chances to obtain a high reliability score. Colluders must choose between colluding for higher reputations or spreading transactions for a higher reliability score.

The most reliable distribution, represented by the 45 degree line, denotes the perfect distribution of transactions among a peer's entire transaction partner set. Application dynamics, however, may result in imperfect distributions resulting in unfairly penalizing honest users. For instance, online storefronts may have a loyal customer base resulting in repeat business from those shoppers. The Gini coefficient could be tuned on a per-application basis such that the *expected* reliability value experienced in the application determines the perfect equality line for the system, and all users are evaluated against the system's average Gini coefficient.

3.3 Addressing unreliable reputations

While our reliability metric allows us to approximate the reliability of a peer's reputation value, it does not provide information about the reliability of the peer itself. A peer can have low-reliability reputations for two reasons: the peer has transacted exclusively with a small number of partners, or it has performed very few transactions, possibly due to having just joined the network. To help assess user reliability

quickly and accurately in high-churn systems, we propose the use of proactive reputations [52].

Where traditional reputation systems rely on feedback given after a transaction, proactive reputations allow a peer to *initiate* transactions with a target for the express purpose of evaluating its reliability for future transactions. For example, take a Peer X that needs to interact with two other peers, A and B, both of whom have unreliable reputations. X can initiate a number of proactive requests to gauge A and B's reliability and trustworthiness. Unlike challenge-response mechanisms where the target has a clear incentive to respond correctly, the goal of proactive requests is to blend in with regular requests to measure the candidate's response to a normal request.

To accurately measure a target peer's true behavior, proactive reputation systems must satisfy several requirements. First, transactions must be relatively low cost to minimize the overhead introduced. Second, they must be verifiable by the sender or a third party to ensure integrity of the feedback. Third, proactive transactions should be anonymous and indistinguishable from normal requests. Request anonymity protects the initiator from detection. Since detection occurs in real-time, we need only a weak level of anonymity which can easily be achieved by redirecting the request through one or more proxy peers. Proxy peers can be trusted third parties or even Sybil accounts belonging to the requester.

When a requester peer, R, uses proactive reputations to test a service provider, P, the result is a set of feedback values that R generates for service provided by P. R can use these results in several ways.

- The proactive feedback values can be added to the pool of transaction feedback values in order to compute a new reputation value, treating them the same as feedback generated by other requesters. This technique, however, is extremely vulnerable to malicious reputation inflation.
- R can compute a local reputation for P based on its first-hand experiences. Given its source, this local reputation value is significantly more reliable than a global reputation. Conservative applications seeking to avoid inflated reputation values can use this value instead of the global reputation.
- Finally, we recommend feedback from proactive reputations be integrated into our reputation reliability metric. Since proactive transactions are forwarded through proxies, P should not be colluding with R. Therefore, we treat each proactive transaction as a transaction with a distinct partner. Thus n proactive transactions with P can be included in our reliability metric as n transactions spread evenly across n unique transaction partners. This maximizes the positive impact on P's reliability metric, but generates a reliability value that is only valid to R. Further, given a desired reputation reliability value, R can compute the number of proactive transactions necessary to reach the target level of reputation reliability. Our evaluations demonstrate that, used in conjunction with proactive reputations, this third technique produces highly accurate measures of user behavior. Our experimental evaluations in the next section demonstrate the effectiveness of using this approach to counter collusion under a dynamic churn environment.

4 Performance evaluation

In this section, we perform detailed evaluation of our proposed solutions and demonstrate their role in improving effectiveness of traditional reputation systems. We begin by discussing our simulation setup, including the peer community, reputation schemes employed, and metrics used to evaluate the reputation mechanisms.

Simulation setup Our experiments are performed on an event-driven network simulator of 5,000 peers. We simulate a large number of peer transactions, where each peer utilizes our reputation framework to choose partners with which to transact. A *transaction* is a two step process: the service requester R, chooses, then performs a transaction with a service provider, P. R then assigns P a binary feedback rating of 0 (negative) or 1 (positive). Our "transactions" are general and effectively represent any type of peer-to-peer requests, including financial transactions, information exchange, file read/write or message forwarding operations.

Before each simulation run, we assign each peer a random *intrinsic trust value* between 0 and 1 that represents the rate at which a peer behaves honestly. For instance, an intrinsic trust value of 0.45 means the peer will provide services or ratings honestly with a probability of 0.45. Since colluders are likely malicious peers with low reliability, we set intrinsic trust values for colluders to random values less than 0.30.

Each experiment run includes two distinct phases: a bootstrap phase and an experiment phase. The bootstrap phase initializes peer reputations for all peers. In this phase, each peer performs transactions with random partners, and rates each provider according to its own intrinsic trust value. We fix the number of bootstrap transactions to 10 in our experiments. We assume that colluders can perform more collusion transactions than regular peers, since collusion transactions often consume less resources. We use our *collusion cost factor* parameter to determine the number of collusion transactions undertaken by colluders during the bootstrap phase. For example, a cost factor of 1:1 means colluders can collude at the same transaction rate as normal network peers, while a cost factor of 5:1 means colluders can perform 5 times as many colluding transactions as normal peers.

Once reputation values have been initialized, we begin our experiment phase. In each run, we conduct 150,000 transactions over 5,000 peers for an average of 30 requests per peer. For each transaction, a peer makes a transaction request, and 25 random peers respond. The initiating peer then uses our reputation framework to choose a transaction partner. We use the partner's intrinsic value to determine if the resulting transaction is a success or failure.

Peer selection algorithms To quantify the benefits of our reputation framework, including the reliability metric and proactive reputations, we compare the performance of three different reputation systems in our experiments: basic reputations (denoted by R), reputations with reliability metric (L), and reputations with reliability metric and proactive reputations (P).

1. *Basic reputations (R):* a peer chooses the service provider with the highest reputation value. We compute Peer i's reputation value, R_i, as the average of all of its past transaction feedback values. Reputations range between 0 and 1.

2. *Reputations with reliability (L):* a peer chooses the provider with the highest weighted combination of reputation and reliability value:

$$L_i = (1 - \alpha) \cdot R_i + \alpha \cdot Q_i. \tag{4}$$

   Q_i, peer i's reliability score, is computed using Eqs. 2 and 3. The weight parameter, α, can be tuned on a per-application basis to favor either higher reputations or more accurate reputations. We fix α to be 0.5 for our experiments.
3. *Reputations with reliability and proactive reputations (P):* Requesters send an average of 5 to 10 proactive probes to providers with reliabilities less than 0.5. As proactive transactions are anonymous, we treat each proactive transaction as a transaction with a distinct user. The resulting firsthand reputation values and transaction histories are then integrated with the globally generated values (Eqs. 3 and 4).

4.1 Effectiveness, accuracy, and overheads

We quantify the impact of our reliability mechanisms using three key metrics: transaction success rate, trust computation error, and metric overheads.

Transaction success rate We measure the rate of successful transactions experienced by all peers in the network. A transaction is deemed successful if the provider behaved honestly. The success rate is the ratio of the number of successful transactions over the total number of transactions in the system, and increases when users can avoid untrustworthy partners by making more accurate trust decisions.

Trust computation error (TCE) This metric represents how accurately a peer's computed reputation reflects its intrinsic trust value. We use our metric as a relative metric to choose between pairs of partners. We define the TCE in terms of a peer's position in an ordered list of peers sorted by computed reputation. For each reputation system, we compute a sorted list of all network peers based on their reputation values. We then compare this ordered list to the sorted list of all peers based on their intrinsic trust values. A peer's TCE is the difference in its position from one list to the other. For example, if, in a network of 10 peers, the most reliable peer (according to intrinsic trust values) has the third highest computed reputation, its per-peer TCE is $(3 - 1)/10$. The TCE of a network is the average TCE of all peers, defined as:

$$TCE = \frac{1}{n} \sum_{k=1}^{n} \frac{|p_c(k) - p_t(k)|}{n}. \tag{5}$$

Here, p_c and p_t respectively refer to positions of peer k's computed trust and intrinsic trust values in the ordered list of all peers sorted on the basis of their reputation values.

Overhead Our reliability metric requires that the network store not only each peer's aggregated trust value, but also a compressed transaction history (in order to compute its reliability value). The transaction history only needs to keep the identity of its past partners and the total number of transactions performed with each partner. We

Table 2 Peer selection algorithms notation

T	Intrinsic trust value
R	Basic reputations
L	Reliability-based reputations
P	Reliability-based and proactive reputations

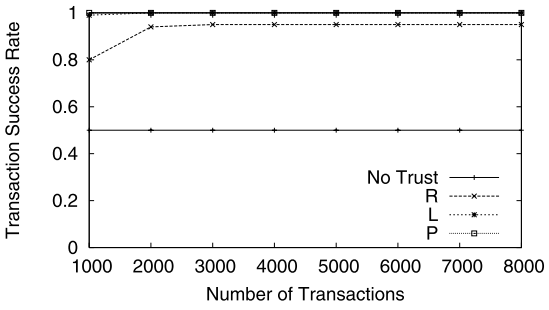

Fig. 7 Benefits of using reputations for peer selection

compute this storage overhead as the number of unique transaction partners per peer. Computational and communication overheads for generating our reliability metric are comparable to a traditional reputation system.

We now present the performance of our reliability mechanism in countering collusion. Each data point represents an average of results from at least three randomized runs.

4.2 Using reputations for peer selection

We first demonstrate the benefit of using reputation-based peer selection in network protocols and applications, as compared to a "random" peer selection scheme. The goal of this simple experiment is to validate previously published benefits of reputations in our experimental context [6, 11, 13, 28, 40, 57]. We simulate a network of 1,000 peers, including 25% dishonest peers that always provide unfair service and unfair ratings to others. We vary the number of transactions, where each transaction is initiated by an honest peer. As Fig. 7 shows, without a trust model, there is a 50% probability of a transaction succeeding. Using a basic reputation model results in a much higher success rate. All reputation-based trust schemes have a success rate close to 100%. Avoiding transactions with these peers reduces the impact of malicious and unreliable network entities, thus improving the robustness of the network protocol or application.

4.3 Resistance to pairwise and Sybil collusion

Pairwise collusion is the most basic form of collusion, where two peers undertake fake transactions to raise each other's reputation. We vary the percentage of pairwise colluders in the network from 10% to 50% on the x-axis, and plot the transaction success rate on the y-axis. As shown in Fig. 8(a), our reliability-based reputations schemes demonstrate a 80% average success rate, and a 30–40% improvement in

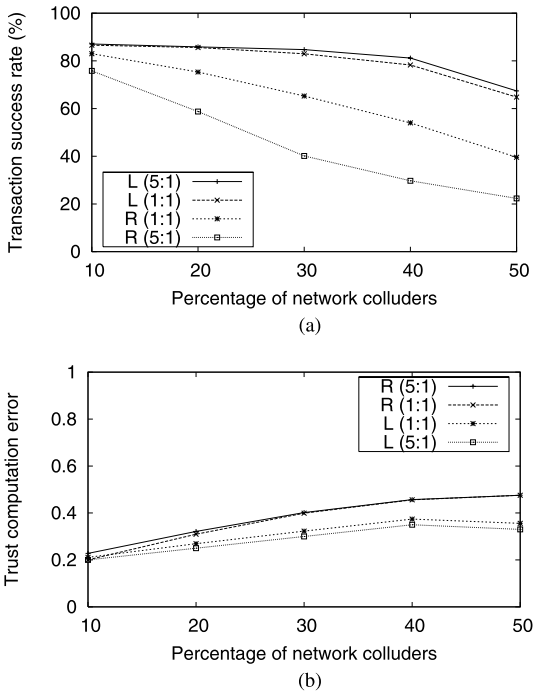

Fig. 8 Effectiveness and accuracy against pairwise collusion. *R* refers to the pure reputations scheme and *L* refers to the reliability-based reputations scheme. The ratios represent the collusion cost factors

network productivity as compared to a pure reputations mechanism. Our mechanism observes little impact with increasing percentage of network colluders. Also, as seen in Fig. 8(b), we observe higher accuracy of peer reputations using our reliability scheme despite the increasing amount of collusion in the network.

Next, we evaluate the effectiveness of reliability-based reputations in countering Sybil colluders. A user launching a Sybil attack can establish multiple slave identities and use them to inflate its reputation. We fix the number of bootstrap transactions to 10 and the number of slaves to 5 per attacker. These slaves only behave as service requesters to the master peer and are not a part of the regular peer community. We vary the percentage of Sybil colluders in the network from 10% to 50% on the x-axis, and plot the transaction success rate on the y-axis. As shown in Fig. 9(a), the pure reputations scheme performs badly with increasing amounts of Sybil attacks and collusion cost ratios in the network. Though we observe a general drop in performance with increasing percentage of users launching these attacks, our mechanism is effective in countering the Sybil attack. We observe a 30% improved success rate even when the Sybils conduct five times as many transactions as normal peers. Similar to our experiment on pairwise collusion, our mechanism observes greater trust computation accuracies as compared to a pure reputations scheme (Fig. 9(b)).

A greater number of colluding slaves helps Sybils inflate their reputations with higher reliability scores (as compared to pairwise colluders). However, transacting with even one non-colluder results in a highly disproportionate Lorenz distribution for the Sybil colluder and a sharp drop in its reliability score. Increasing the magnitude of collusion with each of its slaves further aggravates the poor reliability score of

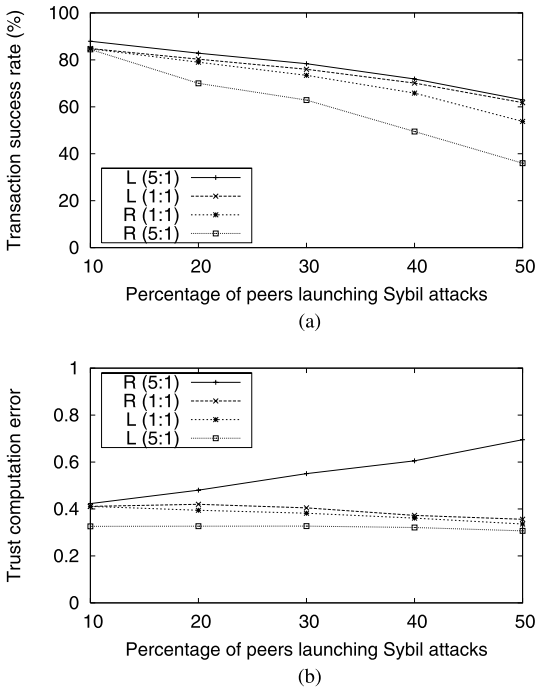

Fig. 9 Effectiveness and accuracy against Sybil collusion. *R* refers to the pure reputations scheme and *L* refers to the reliability-based reputations scheme. The ratios represent the collusion cost factors

the Sybil. A Sybil is challenged to maintain transactions rates per slave comparable to the rates with other non-colluding peers. But this drastically reduces the impact of each colluding partner, resulting in a reputation that more accurately reflects the user's real behavior. We observe similar results for our experiments on the group-based mesh collusion model.

Impact of collusion cost and bootstrap Our next experiment investigates the impact of the amount of pairwise collusion on reputation systems. For this experiment, we fix the number of colluders to 30% of the total population and for each cost factor ratio (1:1 and 5:1), we vary the number of bootstrap transactions undertaken by normal peers (10, 20, and 30 transactions). For example, a bootstrap value of 10 implies that a normal peer conducted 10 bootstrap transactions while a colluder undertook 10 and 50 transactions, respectively, for each cost factor ratio. As shown in Figs. 10(a) and 10(b), an increase in the amount of network collusion results in a drastic drop in performance of the pure reputations scheme. On the other hand, increasing the magnitude of collusion has little to no effect on the success rate of our proposed mechanism. In fact, we observe more accurate results when the amount of pairwise collusion rises in the network, because the inequality in the Lorenz curves for colluders rises sharply when a colluder transacts with even one normal user. Therefore, while these colluders possess high reputations, the reliability of their reputations turns out to be really poor.

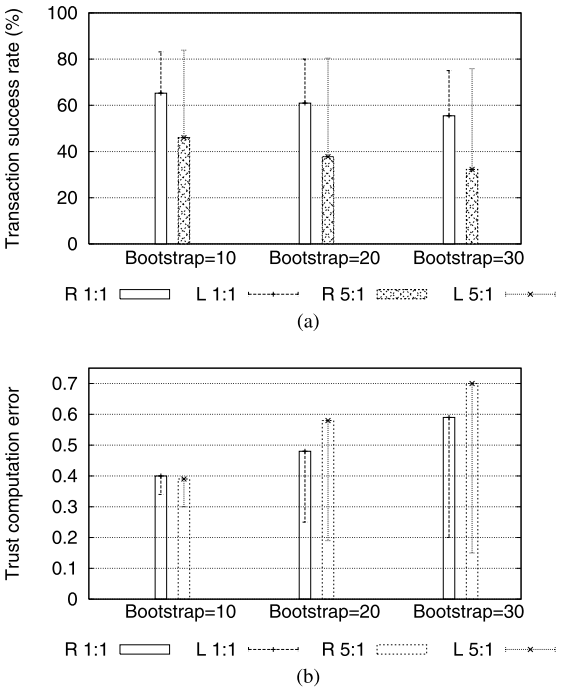

Fig. 10 The effect of collusion cost and bootstrap transactions on effectiveness and accuracy of reliability-based reputations

4.4 Resistance to churn

We employ the Gnutella churn trace to evaluate our proactive reputations scheme. The trace was collected by the University of Washington in May 2001 as part of the Snowtella project. Each node was monitored for approximately 60 hours and the time intervals each node is online in that period is recorded. We limit our experiments to a truncated churn trace of the first 14 hours.

We run our experiments on a simulated community of 5,000 peers with 30% pairwise colluders. Once the bootstrap phase is completed (10 transaction for normal peers, 20 transactions for colluders), we conduct approximately 50,000 transactions over approximately 14 hours, *i.e.*, an average of 10 requests per peer. We conduct one request per time cycle, and choose providers for a transaction based on their availability modeled by the Gnutella churn. That is, for each transaction, we choose a subset of 25 providers that are online at the time the request was made.

Requesters send an average of 5 to 10 proactive probes to providers with reliabilities less than 0.5. As proactive transactions are anonymous, we treat each proactive transaction as a transaction with a distinct user. The consequent firsthand reputation values and transaction histories are then integrated with the globally generated values (as given by Eqs. 3 and 4). The resulting aggregation (P) of proactive reputations and reliability metric provides a flexible and powerful requester-centric perspective of a global reputation.

The objective of this experiment is to observe the error, E, in reputation values as computed using proactive reputations (P) and pure reputations (R) schemes with

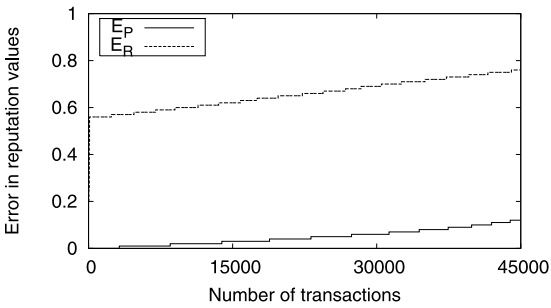

Fig. 11 The performance of proactive reliability-based reputations against Gnutella churn and 30% pairwise collusion. Proactive transactions accounted for less than 5% of all transactions

respect to the intrinsic trust values (T) for each transaction. That is,

$$E_R = |R - T|, \tag{6}$$

$$E_P = |P - T|. \tag{7}$$

Figure 11 illustrates the performance of proactive reputations in the presence of network churn. As shown in the figure, 90% of all transactions experience less than 0.2 error using proactive reputation values while we observe that the pure reputations scheme performs significantly worse with 50% network transactions experiencing greater than 0.6 error in reputation values. Clearly, our reliability-based proactive reputations are very effective even in high churn environments. Additionally, proactive transactions accounted for less than 5% of all transactions in the system. The increased accuracies we observe in our experiment justify this additional proactive traffic.

4.5 Storage overhead

Our objective is to compute additional storage overhead imposed by reliability-based reputations. As part of our performance evaluations, we test our reliability solutions using transaction histories of 18,000 random eBay users.[1] In Fig. 12, we plot the number of unique transaction partners as observed by these sellers. We observe that 60% of the sellers had less than 100 unique transaction partners, and 85% had less than 1,000 unique transaction partners. We believe that storing an average of 100–1000 transaction records per peer is an acceptable overhead given the improved system productivity we observe from our reliability mechanisms.

5 Related work

While P2P networks have been heavily researched, reputation systems are being implemented by practically every Internet application today to improve individual user satisfaction as well as overall system productivity. We first briefly discuss trust-related

[1] Details on the eBay data set are not provided for space reasons. We refer readers to [54] for a complete description of data and other results.

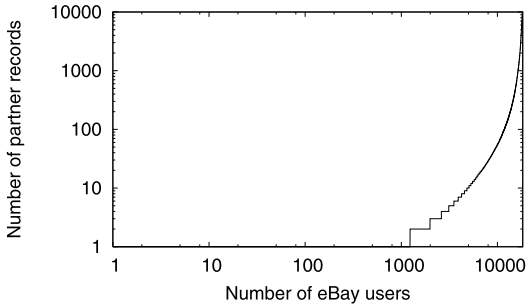

Fig. 12 Storage overhead: the number of unique transaction partner records

research in the context of e-Commerce markets, Web 2.0 applications, wireless networks, and grids. Next, we discuss related surveys in trust and reputations.

5.1 Other applications of reputation systems

e-Commerce marketplaces Reputations today play a fundamental role in the decision to initiate a transaction, and the pricing of goods or services in online marketplaces like eBay.com. eBay's *Feedback Forum* computes a user's reputation as the sum of its lifetime ratings. Reputation profiles are designed to predict future performance and help users decide with who to transact. Sellers with excellent reputations can claim higher prices for their products while poor reputation holders attract fewer buyers [43].

Overstock Auctions is another online auction house similar to eBay in that feedback ratings are aggregated to form a user's "business rating" and "positive percentage" score. Unlike eBay, however, Overstock is unique in its integration of a social network into the market community. Like other social networks, Overstock Auctions encourages users to establish an online presence through personalized homepage with personal history, photos, and links to friends. User profiles often include their shopping preferences and return policies.

Collaborative filtering has witnessed a lot of research because of its high utility in marketplaces like Amazon.com [61]. If A likes gardening and cooking, and B likes gardening, it is likely that B likes cooking too. Collaborative filtering techniques calculate a personalized reputation estimate of an item, X, for a buyer, B, as the weighted average of previous ratings given to X by other buyers. The weights are proportional to the similarity between the buyer, B, and the previous buyers. This buyer similarity is typically calculated as a function of the correlation between ratings assigned to a common set of past sellers. Amazon.com uses such techniques to provide frequent shoppers with personalized recommendations on books, music, and other products that are likely of interest to the user.

Web 2.0 applications Google's PageRank is one of the most popular reputation systems for ranking and web search today [41]. The PageRank algorithm ranks web pages based on the number of links that point to a page as well as the PageRanks of those back links. While extremely successful, the algorithm can be easily manipulated by collusion and Sybil strategies. Zhang et al. observe that colluding nodes

cheat the algorithm by stalling the PageRank random walk in a small web graph, and suggest techniques to capture the amount of PageRank inflation obtained by such collusion [62].

Discussion forms and expert sites like Slashdot, Epinions, and BizRate employ reputation systems to rate experts providing answers and advice in their areas of expertise. Slashdot, for instance, employs an automated moderation mechanism whereby registered users who frequent the site regularly are chosen to either moderate comments to articles, or moderate other moderators.

Lately, online social networks are being explored as an alternative means to establish trust. Social networks have been proposed as a means to defend reputation systems against Sybil attacks [59], and improve the reliability of reputation systems [24, 25]. Kumar et al. study the evolution of structure of the Flickr and Yahoo!360 social networks, and discover that while isolated individuals and communities exist in these networks, there also exists a large strongly connected component spanning the entire network [32]. Others have profiled the power-law, small-world, and scale-free properties of social networks such as Orkut, YouTube, CyWorld, and MySpace [2]. Information from social networks can be exploited to enhance many practical areas of research. Mislove et. al. used social networking data to improve Internet search engines [39], while others have applied this information to increase yields from viral marketing campaigns [15, 46].

Wireless networks and grids Cooperation in wireless ad hoc networks is known to be a critical issue. Significant work in mobile ad hoc networks has gone towards stimulating cooperation and maximizing the throughput of the network [8, 48]. In ad hoc networks, nodes communicate with far off destinations using intermediate nodes as relays. Nodes are energy-constrained and may not always accept relay requests. Nodes could also be malicious and present sophisticated attacks like *denial-of-service* attacks and network *partitioning*. Packets dropped by such free-riding nodes and malicious nodes call for mechanisms to rate nodes and their trustworthiness.

Reputation systems have been employed in wireless ad hoc networks to ensure path reliability and increased throughput in the network [38]. Trust relationships and routing decisions in CONFIDANT, for example, are made based on experienced, observed, or reported routing and forwarding behavior of other nodes [6]. Incentive schemes have been proposed that reward highly reputed nodes with increased traffic in the network [64]. However, most approaches improve cooperation in the network but do not punish maliciousness.

Global public computing systems, called *grids*, also need reputation-based trust mechanisms. Here, clients sometimes pay for services (computing resources) beforehand to untrusted servers. Malicious servers may overcharge clients or dishonor the service agreement. Likewise, malicious clients may refuse to pay or abuse the resources given to them. Reputation schemes have been proposed to reduce free-riding in grids and encourage honesty [18]. Reputations are also suggested as a means to ensure data integrity in Grids where computational results are the primary service products [20].

5.2 Surveys on trust and reputations

Resnick et al. present an excellent introduction to the area of reputation systems [44]. Their work describes three challenges for a successful reputation system: first, entities must be long-lived to account for accurate reputations; second, feedback must be captured and distributed; third, reputations should help distinguish between trustworthy and untrustworthy partners. The area of trust and reputations, in general, has witnessed a tremendous amount of research over the last decade. These efforts, however, have not been systematic and rigorous, partly due to the varied nature of applications and systems that require trust implementations and also due to the innumerable variety of threats that can be posed at any time by malicious entities. There is no single solution suitable in all contexts and applications.

A fair amount of literature, therefore, has surveyed the growth of reputation systems in various contexts like P2P networks [37, 47], Semantic Web [3], multi-agent systems [27], economics and management [19, 43, 45]. Marti and Garcia-Molina present concepts in P2P reputation systems design [37]. While their taxonomy discusses the conflict between system constraints and user behavior, our work presents design issues and threats in reputation systems from the perspective of the four distinct reputation management tasks of collection, aggregation, storage, and communication. Grandison and Sloman classify trust based on the "purpose" it plays in a system like service provision trust (*i.e.*, trust in service or resource provided), access control trust, trust in delegation, identity trust, and so on [22]. Artz and Gil present a review of trust for the Semantic Web, discussing design issues for trust in environments where humans are not the only consumers of information [3]. Josang et al. present the current state-of-the-art in reputations by surveying the reputation systems deployed on the Internet today [27]. Their work also presents a thorough analysis of various reputation aggregation models.

6 Conclusions

Introducing trust and cooperation among strangers is a significant challenge facing an increasing number of large-scale distributed applications. Despite numerous proposals confirming that reputation systems are a powerful mechanism to establish trust, designing a robust and reliable reputation system is still largely an open challenge. The primary goal of this paper has been to investigate and address the critical open challenges that limit the effectiveness of reputations today. We compare prominent reputation schemes proposed by the research community using our taxonomy on reputation management, and examine the extent to which they address the challenges in reputation systems design.

Furthermore, we present solutions to address the two leading reasons for erroneous and misleading values produced by reputation systems today, *i.e.*, user collusion and short-lived online identities. By leveraging the well-accepted Lorenz curve and Gini coefficient, we provide a reliability metric designed to detect and penalize collusion-like behavior, and encourage peers to interact with diverse groups of users across the network. Our evaluations find that our metric complements traditional reputations

well. Together with proactive reputations, an approach that establishes quick and reliable reputations for unknown peers or newcomers, they produce highly accurate measures of user behavior.

References

1. Aberer, K., & Despotovic, Z. (2001). Managing trust in a peer-2-peer information system. In *Proceedings of CIKM*.
2. Ahn, Y. Y., Han, S., Kwak, H., Moon, S., & Jeong, H. (2007). Analysis of topological characteristics of huge online social networking services. In *Proceedings of world wide web (WWW) conference*.
3. Artz, D., & Gil, Y. (2007). A survey of trust in computer science and the semantic web. *Web Semantics: Science, Services and Agents on the World Wide Web*, *5*(2).
4. Bhattacharjee, R., & Goel, A. (2005). Avoiding ballot stuffing in eBay-like reputation systems. In *Proceedings of workshop on economics of peer-to-peer systems (P2PEcon)*.
5. Buchegger, S., & Boudec, J. L. (2001). Nodes bearing grudges: towards routing security, fairness, and robustness in mobile ad hoc networks. In *Proceedings of Euromicro international conference on parallel, distributed and network-based computing (Euromicro-PDP)*.
6. Buchegger, S., & Boudec, J. L. (2004). A robust reputation system for P2P and mobile ad-hoc networks. In *Proceedings of workshop on economics of peer-to-peer systems (P2PEcon)*.
7. Burton, K. (2002). *Design of the openprivacy distributed reputation system*. http://www.peerfear.org/papers/openprivacy-reputation.pdf.
8. Buttyan, L., & Hubaux, J. P. (2003). Stimulating cooperation in self-organizing mobile ad hoc networks. *Mobile Networks and Applications*, *8*(5).
9. Cheng, A., & Friedman, E. (2005). Sybilproof reputation mechanisms. In *Proceedings of workshop on economics of peer-to-peer systems (P2PEcon)*.
10. Dagum, C. (1980). The generation and distribution of income, the Lorenz curve and the Gini ratio. *Economie Appliquée*, 33.
11. Damiani, E., Di Vimercati, D. C., Paraboschi, S., Samarati, P., & Violante, F. (2002). A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of ACM conference on computer and communications security (CCCS)*.
12. Dellarocas, C. (2000). Immunizing online reputation reporting systems against unfair ratings and discriminatory behavior. In *Proceedings of ACM conference on electronic commerce (EC)*.
13. Dewan, P., & Dasgupta, P. (2004). Pride: peer-to-peer reputation infrastructure for decentralized environments. In *Proceedings of world wide web (WWW) conference on alternate track papers and posters*.
14. Dimitriou, T., Karame, G., & Christou, I. (2007). Supertrust: a secure and efficient framework for handling trust in super-peer networks. In *Proceedings of ACM symposium on principles of distributed computing (PODC)*.
15. Domingos, P. (2005). Mining social networks for viral marketing. *IEEE Intelligent Systems*, *20*(1).
16. Douceur, J. (2002). The Sybil attack. In *Proceedings of international workshop on peer-to-peer systems (IPTPS)*.
17. Feldman, M., Lai, K., Stoica, I., & Chuang, J. (2004). Robust incentive techniques for peer-to-peer networks. In *Proceedings of ACM conference on electronic commerce (EC)*.
18. Fernandes, A., Kotsovinos, E., Ostring, S., & Dragovic, B. (2004). Pinocchio: incentives for honest participation in distributed trust management. In *Proceedings of international conference on trust management (iTrust)*.
19. Friedman, E., & Resnick, P. (2001). The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, *10*(2).
20. Gilbert, A., Abraham, A., & Paprzycki, M. (2004). A system for ensuring data integrity in grid environments. In *Proceedings of IEEE international conference on information technology: computers and communications (ITCC)*.

21. Gnutella (2001). *The Gnutella protocol specification v0.4*.
22. Grandison, T., & Sloman, M. (2000). A survey of trust in internet application. *IEEE Communications Surveys and Tutorials*, *4*(4).
23. Hasan, O., Brunie, L., Pierson, J. M., & Bertino, E. (2009). Elimination of subjectivity from trust recommendation. In *Proceedings of the IFIP international conference on trust management*.
24. Hogg, T., & Adamic, L. (2004). Enhancing reputation mechanisms via online social networks. In *Proceedings of ACM conference on electronic commerce (EC)*.
25. Jensen, C., Davis, J., & Farnham, S. (2002). Finding others online: reputation systems for social online spaces. In *Proceedings of ACM SIGCHI*.
26. Josang, A., & Pope, S. (2005). Semantic constraints for trust transitivity. In *Proceedings of the Asia-Pacific conferences on conceptual modelling (APCCM)*.
27. Josang, A., Ismail, R., & Boyd, C. (2007). A survey of trust and reputation systems for online service provision. *Decision Support Systems*, *43*(2).
28. Kamvar, S. D., Schlosser, M. T., & Garcia-Molina, H. (2003). The eigentrust algorithm for reputation management in P2P networks. In *Proceedings of world wide web (WWW) conference*.
29. Kerr, R., & Cohen, R. (2009). An experimental testbed for evaluation of trust and reputation systems. In *Proceedings of the IFIP international conference on trust management*.
30. Kher, V. & Kim, Y. (2005). Securing distributed storage: challenges, techniques, and systems. In *Proceedings of ACM international workshop on storage security and survivability (StorageSS)*.
31. Kollock, P. (1999). The production of trust in online markets. *Advances in Group Processes*, *16*.
32. Kumar, R., Novak, J., & Tomkins, A. (2006). Structure and evolution of online social networks. In *Proceedings of ACM internation conference on knowledge discovery and data mining*.
33. Levien, R. (2000) *Advogato's trust metric*. http://www.advogato.org/trust-metric.html.
34. Lian, Q., et al. (2007). An empirical study of collusion behavior in the maze p2p file-sharing system. In *Proceedings of ICDCS*.
35. Lorenz, M. (1905). Methods for measuring the concentration of wealth. *American Statistical Association*, *9*.
36. Marti, S., & Garcia-Molina, H. (2003). Identity crisis: anonymity vs. reputation in P2P systems. In *Proceedings of P2P*.
37. Marti, S., & Garcia-Molina, H. (2006). Taxonomy of trust: categorizing P2P reputation systems. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, *50*(4).
38. Marti, S., Giuli, T. J., Lai, K., & Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of ACM MobiCom*.
39. Mislove, A., Gummadi, K., & Druschel, P. (2006). Exploiting social networks for Internet search. In *Proceedings of ACM HotNets*.
40. Ooi, B. C., Liau, CY, & Tan, K. L. (2003). Managing trust in peer-to-peer systems using reputation-based techniques. In *Proceedings of the advances in web-age information management (AWAIM)*.
41. Page, L., Brin, S., Motwani, R., & Winograd, T. (1998). *The pagerank citation ranking: bringing order to the web* (Tech. rep.). Stanford Digital Library Technologies Project.
42. Ratnasamy, S., Francis, P., Handley, M., Karp, R., & Schenker, S. (2001). A scalable content-addressable network. In *Proceedings of ACM SIGCOMM*.
43. Resnick, P., & Zeckhauser, R. (2001). Trust among strangers in internet transactions: empirical analysis of eBay's reputation system. *Advances in Applied Microeconomics*, *11*.
44. Resnick, P., Kuwabara, K., Zeckhauser, R., & Friedman, E. (2000). Reputation systems. *Communications of the ACM*, *43*(12).
45. Resnick, P., Zeckhauser, R., Swanson, J., & Lockwood, K. (2006). The value of reputation on eBay: a controlled experiment. *Experimental Economics*, *9*(2).
46. Richardson, M., & Domingos, P. (2002). Mining knowledge-sharing sites for viral marketing. In *Proceedings of ACM internation conference on knowledge discovery and data mining*.
47. Ruohomaa, S., Kutvonen, L., & Koutrouli, E. (2007). Reputation management survey. In *Proceedings of IEEE international conference on availability, reliability and security (ARES)*.
48. Srinivasan, V., Nuggehalli, P., Chiasserini, C., & Rao, R. (2003). Cooperation in wireless ad hoc networks. In *Proceedings of IEEE INFOCOM*.
49. Srivatsa, M., Xiong, L., & Liu, L. (2005). Trustguard: countering vulnerabilities in reputation management for decentralized overlay networks. In *Proceedings of world wide web (WWW) conference*.
50. Stoica, I., Morris, R., Karger, D., Kaashoek, M. F., & Balakrishnan, H. (2001). Chord: a scalable peer-to-peer lookup service for internet applications. In *Proceedings of ACM SIGCOMM*.
51. Swamynathan, G., Zhao, B., & Almeroth, K. (2006). Exploring the feasibility of proactive reputations. In *Proceedings of international workshop on peer-to-peer systems (IPTPS)*.

52. Swamynathan, G., Zhao, B., & Almeroth, K. (2007). Exploring the feasibility of proactive reputations. *Concurrency and Computation: Practice and Experience, Special Issue on Recent Advances in P2P Systems and Security*, *20*(2).
53. Swamynathan, G., Zhao, B., Almeroth, K., & Zheng, H. (2007). Globally decoupled reputations for large distributed networks. *Advances in Multimedia*, *2007*(1).
54. Swamynathan, G., Zhao, B., Almeroth, K., & Jammalamadaka, S. R. (2008). Towards reliable reputations for dynamic networked systems. In *Proceedings of IEEE international symposium on reliable distributed systems (SRDS)*.
55. Symantec (2000). *Vbs.Gnutella worm*. http://securityresponse.symantec.com/avcenter/venc/data/vbs.gnutella.html.
56. Walsh, K., & Sirer, E. G. (2006). Experience with an object reputation system for peer-to-peer file-sharing. In *Proceedings of Usenix networked systems design and implementation (NSDI)*.
57. Xiong, L., & Liu, L. (2004). Peertrust: supporting reputation-based trust for peer-to-peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, *16*(7).
58. Yang, M., Chen, H., Zhao, B. Y., Dai, Y., & Zhang, Z. (2004). Deployment of a large-scale peer-to-peer social network. In *Proceedings of Usenix workshop on real, large distributed systems (WORLDS)*.
59. Yu, H., Kaminsky, M., Gibbons, P. B., & Flaxman, A. (2006). Sybilguard: defending against Sybil attacks via social networks. In *Proceedings of ACM SIGCOMM*.
60. Yu, H., Gibbons, P. B., Kaminsky, M., & Xiao, F. (2008). Sybillimit: a near-optimal social network defense against Sybil attacks. In *IEEE symposium on security and privacy*.
61. Zacharia, G., Moukas, A., & Maes, P. (2000). Collaborative reputation mechanisms for electronic marketplaces. *Decision Support Systems*, *29*(4).
62. Zhang, H., Goel, A., Govindan, R., Mason, K., & Roy, B. V. (2004). Making eigenvector-based reputation systems robust to collusion. In *Proceedings of the international workshop on algorithms and models for the web-grap (WAW)*.
63. Zhao, B. Y., Huang, L., Rhea, S. C., Stribling, J., Joseph, A. D., & Kubiatowicz, J. D. (2004). Tapestry: a global-scale overlay for rapid service deployment. *IEEE Journal on Selected Areas in Communications*, *22*(1).
64. Zhong, S., Chen, J., & Yang, Y. R. (2003). Sprite: a simple, cheat-proof, credit-based system for mobile ad-hoc networks. In *Proceedings of IEEE INFOCOM*.
65. Zhou, R., & Hwang, K. (2007). Powertrust: a robust and scalable reputation system for trusted peer-to-peer computing. *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, *18*(4).