SocialTrust: Tamper-Resilient Trust Establishment in Online Communities

James Caverlee
Dep't of Computer Science
Texas A&M University
College Station, TX 77843
caverlee@cs.tamu.edu

Ling Liu
College of Computing
Georgia Tech
Atlanta, GA 30332
lingliu@cc.gatech.edu

Steve Webb
College of Computing
Georgia Tech
Atlanta, GA 30332
webb@cc.gatech.edu

ABSTRACT

Web 2.0 promises rich opportunities for information sharing, electronic commerce, and new modes of social interaction, all centered around the "social Web" of user-contributed content, social annotations, and person-to-person social connections. But the increasing reliance on this "social Web" also places individuals and their computer systems at risk, creating opportunities for malicious participants to exploit the tight social fabric of these networks. With these problems in mind, we propose the SocialTrust framework for tamper-resilient trust establishment in online communities. SocialTrust provides community users with dynamic trust values by (i) distinguishing relationship quality from trust; (ii) incorporating a personalized feedback mechanism for adapting as the community evolves; and (iii) tracking user behavior. We experimentally evaluate the SocialTrust framework using real online social networking data consisting of millions of MySpace profiles and relationships. We find that SocialTrust supports robust trust establishment even in the presence of large-scale collusion by malicious participants.

Categories and Subject Descriptors: H.3.5 Information Storage and Retrieval: Online Information Services

General Terms: Algorithms, Experimentation

1. INTRODUCTION

The past few years have seen the explosive rise of Web-based social networks (like Facebook and MySpace), online social media sites (like YouTube, Digg, and Flickr), and large-scale information sharing communities (like Wikipedia and Yahoo! Answers) – all part of a Web 2.0 push that has attracted increasing media, industry, and research interest. One of the cornerstones of Web 2.0 is social or community-based information management, for enhancing the quality of traditional Web search and information retrieval approaches by leveraging the inherent social connections between users and other users via social networks, social tagging, and other community-based features (e.g., social collective intelligence) [4, 7, 20].

But these opportunities have not come without a price, as malicious participants are increasingly targeting these communities in an effort to exploit the perceived social bonds inherent in community-based information management. For example, malicious users can exploit the perceived social connection between users for increasing the probability of disseminating misinformation, of driving participants to the seedy side of the Internet (e.g., to sites hosting malware), and of other disruptions to the quality of community-based knowledge. Some of these risks have already been observed in existing Web 2.0 applications, including impersonated (or fraudulent) digital identities [30], targeted malware dissemination [6], social network enhanced phishing [15], and corrupt user-generated metadata (or tags) [18]. Detecting and mitigating this social spam and social deception is challenging, especially as adversaries continue to adapt their tactics and strategies.

With these problems in mind, we focus on building an online community platform that allows wide access to many different types of users and that still remains useful, even in the presence of users intent on manipulating the system. As a first step towards this goal, we propose SocialTrust, a reputation-based trust aggregation framework for supporting tamper-resilient trust establishment in online communities. The benefits of reputation-based trust from a user's perspective include the ability to rate neighbors, a mechanism to reach out to the rest of the community, and some assurances on the trustworthiness of unknown users in the community. Reputation systems are an important feature of many e-marketplaces and online communities (like eBay, Amazon, and Digg), and reputation-based trust systems have received considerable attention in P2P systems (e.g., [1, 17, 22]). Most existing approaches, however, ignore the social constructs and social network topology inherent in online communities, and typically provide less personalized criterion for providing feedback and computing reputations.

A key challenge then is whether we can develop a trust model for online communities that is tamper-resilient even in the presence of malicious users. And what are the critical factors impacting such a model? We believe that understanding the dynamics of trust establishment can have wide-ranging impact in large-scale collaborative digital libraries, in question answering communities like Yahoo! Answers, in Wikipedia-style information sharing communities, and in other community-based information management systems.

In particular, we carefully consider the unique proper-

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
JCDL'08, June 16–20, 2008, Pittsburgh, Pennsylvania, USA.
Copyright 2008 ACM 978-1-59593-998-2/08/06 ...$5.00.

ties of social networks to build the SocialTrust model for tamper-resilient trust establishment in online communities. Three of the salient features of SocialTrust are:

- *Distinguishing relationship quality from trust* – Many trust approaches make no distinction between the trust placed in a user and the trust placed in a user's relationships. SocialTrust incorporates these distinct features, leading to better resistance to trust manipulation.
- *Personalized feedback* – SocialTrust augments the relationships in the social network with a personalized feedback mechanism so that trust ratings may be dynamically updated as the social network evolves.
- *Tracking user behavior* – SocialTrust incorporates the evolution and trajectory of a user's trust rating to incent long-term good behavior and to penalize users who build up a good trust rating and suddenly "defect."

We experimentally evaluate the SocialTrust framework over a simulated information sharing community based on real social network data consisting of millions of MySpace profiles and relationships. While other trust aggregation approaches have been developed and implemented by others, we note that it is rare to find such a large-scale experimental evaluation. We find that in the context of large-scale attempts to undermine the quality of ratings that it is significantly more robust than popular alternative trust models.

2. RELATED WORK

The study of social networks has a rich history [24], and there has been great interest in modeling these networks and understanding how people efficiently use their social networks, e.g., [12, 32, 33]. The rise of online communities has spurred interest in community information management [11], social network formation [3], and the modeling and analysis of online social networks [2, 19, 21].

In this paper we consider the problem of *trust aggregation in online communities*, which can build on previous work on the important, but distinct, problem of assessing *direct trust*. Several studies have examined how to compute direct trust between nodes, including: [29], which developed statistical models of bid behavior on eBay to determine which sellers are suspicious; TrustGuard [31], which targeted strategically malicious P2P nodes who oscillate their behavior; PeerTrust[35], which studied P2P feedback mechanisms; [23], which stresses the need for direct trust; [13], which studies personalized trust and distrust propagation; and [37], which studied reputation formation in electronic communities. Note that direct trust can be interpreted differently depending on the context and the relevant community: for example, in eBay, trust is a measure of the fulfillment of a commitment; in a P2P network, trust is often a measure of file download success. Our goal in this paper is to propose a general framework for trust aggregation that can incorporate any of these direct approaches; in fact, elements of each of these direct trust approaches can be layered into the SocialTrust approach (see Section 4 below). Experimentally, we do ground our evaluation in the specific context of community-based information sharing.

Research on trust and reputation in P2P networks (e.g., [1, 5, 10, 17, 22]) and on the Web (e.g., [14, 34]) can inform the development of SocialTrust. Note that there are some key differences between these environments and social networks. For example, P2P networks often are concerned with high node churn and guaranteeing anonymity, and the networks are often formed via randomization protocols for establishing links between nodes. In contrast, online social networks tend to include long-lived profiles that strive to be known (i.e., are not anonymous), and links in the social network stress the personal connection. On the Web, users can rely on trust ratings over pages; typically, the user is divorced from the page-level trust assessment which often centers around hyperlink analysis. On social networks and in SocialTrust in particular, users are first-class participants in how trust is built and used.

3. THE SOCIALTRUST MODEL: OVERVIEW

In this section, we introduce the overall SocialTrust model. The goal of SocialTrust is to enhance online communities by providing a trust rating for each user. A trust-based approach is one of the most promising avenues for maintaining the relative openness of these communities (and the corresponding benefits) and still providing some measure of resilience to vulnerabilities. Using trust ratings, a user can decide with whom to engage in new social interactions, to form new groups, to engage in transactions, and so on.

3.1 Reference Model

We model an online social network $\mathcal{SN}$ as a triple consisting of profiles $\mathcal{P}$, relationships $\mathcal{R}$, and contexts $\mathcal{C}$: $\mathcal{SN} = <\mathcal{P}, \mathcal{R}, \mathcal{C}>$. A profile p is the online representation of a particular person, place, or thing. Typically, a profile is a user-controlled Web page that includes some descriptive information about the person it represents. We denote the set of all profiles in the social network $\mathcal{SN}$ as $\mathcal{P}$. We shall assume there are n profiles in the network, numbered from 1 to n: $\mathcal{P} = \{p_1, ..., p_n\}$. We denote a relationship between profiles i and j with two entries in the relationship set $\mathcal{R}$ to characterize each participant's contextual view of the relationship: $rel(i, j, c_1)$ and $rel(j, i, c_2)$, where c_1 and c_2 are two contexts drawn from the context set $\mathcal{C}$. We denote user i's set of contacts as $rel(i)$ and the total number of relationships i participates in as $|rel(i)|$. A relationship in a social network is a bidirectional link between two users. A relationship is only established after both parties acknowledge the relationship. The context indicates the nature of the relationship – e.g., the two people are co-workers.

3.2 Assessing Trust with SocialTrust

We denote the SocialTrust trust rating of user i at time t by $ST(i, t)$. For any two users in the community, we may evaluate the relative trustworthiness, e.g., that user i is more trustworthy than user j (i.e., $ST(i, t) > ST(j, t)$). This aggregated trust information may be used by users for enhancing the quality of their experiences in the community. Since users will typically have direct relationships with only a small fraction of all users in the network, trust values may be used to evaluate the quality of the vast majority of other users for which the user has no direct experience.

For presentation clarity, we shall assume the presence of a centralized *trust manager* whose job is to compute trust ratings for users in the network and to communicate these trust ratings to users when needed. Alternatively, the duties of the trust manager may be securely distributed throughout the network (see, for example, [16]).

Initially all users are treated equally. SocialTrust supports trust maintenance through dynamic revision of trust ratings according to three critical components: the current quality component of trust $Tr_q(i,t)$, the history component, and the adaptation to change component.

$$ST(i,t) = \alpha \cdot Tr_q(i,t) + \beta \cdot \frac{1}{t}\int_0^t ST(i,x)dx + \gamma \cdot Tr'_q(i,t) \quad (1)$$

where $Tr'_q(i,t)$ is the derivative of $Tr_q(i,x)$ at $x = t$. This approach is similar to a Proportional-Integral-Derivative (PID) controller used in feedback control systems [26].

- **Quality Component of Trust** $[Tr_q(i,t)]$: The first component of SocialTrust is the quality component $Tr_q(i,t)$ which provides a snapshot of the trustworthiness of the user based on the current state of the social network. Developing a high-quality core trust metric is very important, and so we shall study this component in great detail in the following section.
- **History Component of Trust** $[\frac{1}{t}\int_0^t ST(i,x)dx]$: The second component considers the evolution of a user's trust rating. This history component is important for (i) providing an incentive to all users in the network to behave well over time; and (ii) limiting the ability of malicious participants to whitewash their trust ratings by repeatedly leaving and re-entering the network.
- **Adaptation to Change Component of Trust** $[Tr'_q(i,t)]$: The final SocialTrust component tracks shifts in a user's behavior. This change component can mitigate the impact of malicious participants who build up a good trust rating over time (through the other two components) and suddenly "defect."

The overall SocialTrust approach is unique on at least two counts. First, most existing trust research ignores trust history and change adaptation, even though it is clear that these are critical factors to ensure quality trust ratings over time. Second, the core SocialTrust metric $Tr_q(i,t)$ distinguishes relationship quality from trust and supports personalized feedback through personalized trust group formation as we shall discuss in Section 4 below.

By tuning α, β, and γ, the SocialTrust model can be optimized along a number of dimensions, e.g., (i) to emphasize the most recent behavior of a user in the network (by choosing higher values of α); (ii) to de-emphasize the current user's behavior in the context of his entire history of behavior (by choosing higher values of β); or (iii) to amplify sudden fluctuations in behavior (by choosing higher values of γ). In addition, the history and change adaptation components of trust allow the overall SocialTrust rating to tolerate errors in the calculation of the node's current trust rating ($Tr_q(i,t)$). In practice, the appropriate setting for these tunable knobs is application and scenario dependent. In our ongoing research, we are deploying SocialTrust internally at Texas A&M to study these choices in more detail.

We shall focus the rest of this paper exclusively on how to compute the base trust metric $Tr_q(i,t)$. Returning to Equation 1, we can see how the overall SocialTrust approach is a function of this base trust metric, its derivative, and a history of previous SocialTrust scores. Hence, the quality of SocialTrust relies heavily on the choice of a good base trust metric. For clarity, we shall drop the time subscript, and refer to the trust score for user i as $Tr_q(i)$.

4. SOCIALTRUST: CORE TRUST MODEL

In this section, we discuss how to compute the quality component of each user's overall trust rating $Tr_q(i)$ through an analysis of the relationships in the social network $\mathcal{SN}$. We view the social network $\mathcal{SN} =< \mathcal{P}, \mathcal{R}, \mathcal{C} >$ as a graph where the profiles $\mathcal{P}$ are nodes and the relationships $\mathcal{R}$ are labeled directed edges. A node in the graph represents one profile. A labeled directed edge in the graph represents a relationship link from one profile to another. A relationship link from profile i to j is represented by the edge from node i to node j in the graph and is labeled with a context c. We treat relationships maintained by each user as a list of recommendations of other users, and view the directed graph $\mathcal{SN}$ as a *recommendation-based trust network*, where a relationship link from user i to user j is treated as a recommendation by user i of user j. Based on this recommendation structure, we develop the SocialTrust quality component of trust.[1]

4.1 Preliminary Trust Models

We begin our development of the SocialTrust quality component by considering a basic trust model that considers the sheer quantity of recommendations for evaluating the trustworthiness of participants:

$$Tr_q(i) = |rel(i)| \quad (2)$$

This recommendation count has close analogs in other network analysis domains, including bibliometrics and traditional social network analysis (where popularity can be measured by a count of contacts or friends) [25]. The basic popularity trust model is subject to extreme manipulation by malicious (or even just ego-centric) participants, especially since online identities are cheap (often requiring only a valid email address for authentication).

A natural extension of the basic popularity trust model is to consider both the number of recommendations for a user *and* the quality of the users making the recommendations. This trust formulation can be written in a recursive fashion:

$$Tr_q(i) = \sum_{j \in rel(i)} Tr_q(j)/|rel(j)| \quad (3)$$

Equivalently, this approach can be described in terms of a random walker who behaves in a manner similar to the random surfer of the popular PageRank approach for Web ranking [27]. The random walker proceeds across the recommendation network; at each node i, the random walker follows one of i's recommendation links with probability $1/|rel(j)|$. In the long run, the random walker will visit high-quality users more often than low-quality ones.

Such random walk models have been studied in both the peer-to-peer file-sharing domain [17] and in the context of trust management for the Semantic Web [28]. More recently, similar random walk models have been applied to social networks (where nodes are users and links are the relationships between users) in [36] and studied more closely in the context of expertise networks in [38].

[1] Whether all relationships can be rightly treated as recommendations is an open question and beyond the scope of this paper. We emphasize that we make no requirements on how relationships in the community arise, and all of the algorithms presented in this paper are agnostic to this relationship formation.

4.2 The Core Trust Model

Two key observations motivate our current work:

1. Distinguishing Relationship Quality from Trust. Many trust models (e.g., [14, 17]) evaluate the relative trustworthiness of a node (or user, in our case) based on the trustworthiness of all nodes pointing to it, but make no distinction about the relationship (or link) quality of each node. In essence, these approaches make no distinction between the trust placed in a user and the trust placed in a user's relationships. Intuitively, we would like to differentiate between users who consistently engage in high-quality relationships with other users versus users who tend to engage in lower quality relationships.

2. Incorporating Personalized User Feedback. Second, trust models based solely on network topology are divorced from the underlying behavior of the users in the network. Relationships in the online social network provide the basis for trust aggregation, but there is no feedback mechanism for dynamically updating the quality of the trust assessments based on how well each user in the network behaves. Hence, we are interested in "closing the loop" so that the trust assessments may be dynamically updated as the social network evolves and as the quality of each user (with respect to user feedback) changes over time.

Hence, the core SocialTrust trust metric considers three key factors to support trust establishment:

- **Trust Establishment Scope:** The trust establishment scope governs which other participants in the social network a user can make an assessment of (and which other participants can make an assessment of that user).
- **Trust Group Feedback:** The second key component of trust establishment is the feedback rating of participants in the network. User i's feedback rating $F(i)$ could be used directly for trust establishment, but it takes no advantage of the rich social connections of the online social network for evaluating user trustworthiness.
- **Relationship Link Quality:** Hence, the third component of trust establishment for a user in the social network is that user's relationship link quality, denoted $L(i)$. One of the benefits of link quality is that it provides an incentive for users to monitor the quality of their relationships.

The core SocialTrust trust metric incorporates the scope, feedback information, and relationship link quality into the trust assessment of each user for improving the quality and robustness of the trust assessments. The intuition is that a user's trustworthiness should be determined by both: (i) the number and trustworthiness of the users who recommend her; and (ii) the relationship link quality of each recommending user. In this way, a recommendation from a high-trust/high-link-quality user counts more than a recommendation from a high-trust/low-link-quality user.

$$Tr_q(i) = \sum_{j \in rel(i)} L(j) \cdot Tr_q(j)/|rel(j)| \qquad (4)$$

This formula states that the trustworthiness of user i is determined by the trustworthiness $Tr_q(j)$ and the link quality $L(j)$ of the users that recommend her, as well as by the number of recommendations made by user j (via the factor $|rel(j)|$).[2] In this sense, the recommendation weights are used to determine how a user's "vote" is split among the users that it recommends, but the relationship link quality of a user impacts how large or small is the user's vote.

To incorporate feedback ratings, we can augment Equation 4 to arrive at the final SocialTrust trust metric:

$$Tr_q(i) = \lambda \sum_{j \in rel(i)} L(j) \cdot Tr_q(j)/|rel(j)| + (1-\lambda)F(i) \qquad (5)$$

where the feedback rating $F(i)$ favors users who have been rated highly by other users within the trust establishment scope, according to the mixing factor $1-\lambda$. This final SocialTrust trust assessment incorporates the relationship structure of the social network, feedback ratings, trust establishment scope, and relationship link quality to provide a global trust value to each user in the social network.

Given the core SocialTrust trust metric, there are a number of open questions. How is the trust establishment scope formed? How do we aggregate user feedback? How is relationship quality assessed? What are the important factors impacting robust trust assessment? In the following sections, we address each of these questions in turn to provide a thorough understanding of SocialTrust and how it supports robust trust establishment.

4.3 Trust Establishment Scope

The trust establishment scope governs what other participants in the network each user can judge, and what other participants can judge each user. Trust group formation can be tuned to balance efficiency and the security of the overall system (by constraining users from manipulating the reputation of users outside of their trust group). At one extreme, there is a single *trust group* consisting of all members of the social network. At the other extreme, each user belongs to a lone trust group consisting of only themselves, meaning that the system supports no trust aggregation. For balancing these two extremes, we could rely on trust groups defined by self-described interests (e.g., sports), location (e.g., members who all live in Texas), or other contextual information.

In this paper, we propose to define trust groups based on the chains of relationships that are fundamental to the formation of social networks. Hence, we consider a *relationship-based* model for determining a user's trust group where the size of the trust group is determined by a network-specified *radius*, ranging from a user's direct neighbors (radius 1), to a user's direct neighbors plus his neighbors' neighbors (radius 2), and so on. By limiting the radius of a user's trust group, we can constrain the impact of malicious users who are far away in the social network. In practice, we form relationship-based trust groups through the *browse-based search capability* provided by most online social networks, whereby a user's profile may be viewed (or browsed) by other users. Users may manually browse from profile to profile and provide ratings to the trust manager on users encountered subject to the radius of the relationship-based trust group.

4.4 Assessing Trust Group Feedback

Given a trust group, we next describe several strategies for assessing the trust group feedback in SocialTrust. We assume that each user i in the network is associated with a

[2] Contextual information (recall the context set $\mathcal{C}$) can be used to revise this uniform split, for example, to favor recommendations from friends and family over recommendations from co-workers.

feedback value $F(i)$ that indicates how well the user's trust group views the user. The feedback ratings are taken from the interval $[0, 1]$. We make two observations: (i) user behavior is dynamic, so the feedback ratings should be dynamically updated; and (ii) malicious users may attempt to subvert them.

For assessing feedback ratings, each user maintains state about the other users it has made a rating for through browse-based search. Based on the ratings of all users who have interacted with user j, we can assess a feedback rating $F(j)$. Guaranteeing that feedback ratings are robust to manipulation is an important feature, and there have been several recent studies on how to ensure such robustness (e.g., [29, 31, 35]) in addition to securing the voting infrastructure (e.g., through encrypted votes, secure transmission, etc.).

In this paper, we rely on a fairly basic rating scheme to show the power of the SocialTrust framework even without these more sophisticated techniques; we anticipate revisiting this issue in future work. A vote is a pair of the form $< user, vote >$, where *user* is a unique user identifier (the profile number) and *vote* is either "good" or "bad". Each user communicates to the trust manager a vote for user it has interacted with in the most recent period. We consider three voting schemes – (i) open voting; (ii) restricted voting; and (iii) trust-aware restricted voting. We describe the first two and their drawbacks to motivate the final trust-aware restricted voting scheme.

Open Voting: We use the shorthand $v_i(j)^+$ to indicate a "good" vote by user i for user j; $v_i(j)^-$ indicates a "bad" vote. In the simplest case user j's feedback rating $F(j)$ is the fraction of "good" votes cast for user j:

$$F(j) = \frac{\sum_i \mathcal{I}(v_i(j)^+)}{\sum_i \mathcal{I}(v_i(j)^+) + \mathcal{I}(v_i(j)^-)}$$

where the indicator function $\mathcal{I}(\cdot)$ resolves to 1 if the argument to the function is true, and 0 otherwise. This open voting policy is subject to ballot stuffing. A single malicious user can issue an unlimited number of "good" votes for raising the feedback rating of colluding users or can issue "bad" votes for demoting the feedback rating of competing users.

Restricted Voting: We can restrict how much each user can vote by assigning each user a limited number of *points* to be allocated over all of its votes. We let w_{ij} denote the number of points user i uses to weight her vote for user j, where the total points allocated to each user is an arbitrary constant: $\sum_j w_{ij} = 1$. Hence, this restricted voting leads to a new feedback rating:

$$F(j) = \frac{\sum_i w_{ij}\mathcal{I}(v_i(j)^+)}{\sum_i w_{ij}\mathcal{I}(v_i(j)^+) + w_{ij}\mathcal{I}(v_i(j)^-)}$$

The trust manager will only accept up to $\sum_j w_{ij} = 1$ points per voter i. All votes over the restriction will be ignored. By restricting the total size of vote allocated to each user, this restricted voting scheme avoids the problem of vote stuffing by a single user. We have no assurances that a malicious user will choose to vote truthfully for other users it has actually interacted with, but we do know that the total amount of voter fraud is constrained. Unfortunately, such a voting scheme is subject to collusive vote stuffing, in which many malicious users collectively decide to boost or demote the feedback rating of a selected user.

Trust-Aware Restricted Voting: To handle the problem of collusive vote stuffing, we advocate a weighted voting scheme in which users are allocated voting points based on how trustworthy they are. We again let w_{ij} denote the number of points user i uses to weight her vote for user j, but now the total points allocated to each user depends on her trustworthiness: $\sum_j w_{ij} = ST(i)$. This trust-aware restricted voting scheme results in a feedback rating for user j of:

$$F(j) = \frac{\sum_i ST(i) w_{ij}\mathcal{I}(v_i(j)^+)}{\sum_i ST(i) w_{ij}\mathcal{I}(v_i(j)^+) + ST(i) w_{ij}\mathcal{I}(v_i(j)^-)}$$

The trust manager will only accept up to $\sum_j w_{ij} = ST(i)$ points per voter i. All votes over the restriction will be ignored, meaning a malicious user cannot ballot stuff. If a malicious user receives poor feedback from trusted users in the system, then his feedback rating will be negatively affected, which in turn will impact his trustworthiness in the system. Intuitively, this cycle is appealing since it can dynamically adapt to trusted users who over time begin behaving badly as well. Note that other feedback approaches are possible and easily pluggable into the SocialTrust framework.

4.5 Assessing Relationship Quality

In this section, we discuss the third critical factor of the core SocialTrust metric – relationship link quality. Recall that user i participates in a total number of relationships $|rel(i)|$. How many of these relationships are with high quality users? Our goal in this section is to formally assess the quality of a user's relationship links. Concretely, let $L(i)$ denote the *relationship link quality* of user i. A score of $L(i) = 0$ indicates that user i has poor quality relationship links. In contrast, a score of $L(i) = 1$ indicates that user i has high quality relationship links.

The small world nature of many social networks means that a large portion of the network may be reachable from any one user within a few hops. Hence, a user's relationship link quality should depend on the user's direct relationship links and perhaps the relationship links of its neighbors up to some small number (k) of hops away. We also observe that a user's link quality should be related to the feedback ratings of its neighbors. A user who only engages in relationships with well-behaving users should earn a higher link-quality score than a user who has relationships with poorly behaving members of the network. We next formally define link quality and provide a discussion of the factors impacting its assessment.

4.5.1 Link Quality as a Scoped Random Walk

We model the link quality of user i in terms of a scoped random walk model, in which a random walker originates its walk at user i and randomly follows the relationship links of user i and the subsequent users at which it arrives up to some small number of steps.

In the extreme, when all users within k hops of the original user i have a perfect feedback rating (i.e., $F(j) = 1$ for all users within k hops of i), then user i has link quality $L_k(i) = 1$. In contrast, if user i either has a poor feedback rating (i.e., $F(i) = 0$) or all of the users within k hops of user i have poor feedback ratings, then user i's link quality is $L_k(i) = 0$. To summarize, the link quality of user i can be interpreted as the probability that a random walker originating its walk at user i ends at a high-quality user after walking up to k-hops away from i.

We can begin our examination of link quality by considering the base case when the scope (k) is 0.

Base Case (k=0): In the base case, the link quality of a user is merely its feedback rating $F(i)$:

$$L_{[0]}(i) = F(i)$$

The random walker walks for 0-hops, meaning that it stays at the original user. The probability that the random walker ends at a high-quality user is thus $F(i)$.

One-Hop Case (k=1): In the one-hop case, the link quality of a user is the probability that the random walker ends at a high-quality user after walking forward to one of the contacts from the original user's set of relationships (recall that $rel(i)$ denotes the relationship list for user i):

$$L_{[1]}(i) = F(i) \sum_{j \in rel(i)} F(j)/|rel(i)|$$

Note that the random walker proceeds initially according to the feedback rating $F(i)$ of the original user. Accordingly, the link quality of a user that has received poor feedback will be low. But a user with a high feedback rating who recommends poor quality users will also be penalized with a low link quality.

Two-Hop Case (k=2): The link quality can be extended to consider random walks of length two, where:

$$L_{[2]}(i) = F(i) \sum_{j \in rel(i)} F(j)/|rel(i)| \left[\sum_{l \in rel(j)} F(l)/|rel(j)| \right]$$

We can extend link quality to consider random walks of arbitrary length k. In all cases, link quality is a local computation and can be updated in a straightforward fashion.

4.5.2 Correction Factor

The scoped random walk provides a natural measure of the relationship link quality of each user. However, the feedback ratings used for driving the link quality assessment may not be known with certainty and malicious users may attempt to subvert these ratings (recall Section 4.4). Hence, in this section, we discuss several correction factors for augmenting the basic scoped random walk model in the presence of such uncertainty. We denote the updated link quality score for user i as $\hat{L}_{[k]}(i)$, and evaluate it in terms of the original link quality score and a correction factor ϕ:

$$\hat{L}_{[k]}(i) = \phi \cdot L_{[k]}(i)$$

We present an optimistic and a pessimistic correction factor as two baseline approaches to motivate a hop-based correction factor. The hop-based factor balances the extremes of and pessimistic factors for guiding the proper link quality correction factor for each user.

Optimistic Correction: The optimistic correction factor makes no changes to the original link quality as determined by the scoped random walk. For all users, the optimistic correction factor is 1:

$$\phi_{opt}(i) = 1, \forall i$$

The optimistic approach will tend to over-estimate the link quality of users that (i) are part of a malicious clique in which some users behave well to mask their relationships with clique members who behave poorly; or (ii) engage in relationships with poor quality users for whom the feedback ratings have incorrectly identified as high quality.

Pessimistic Correction: The pessimistic correction factor treats a user with even a very small likelihood (call it δ) of recommending a poorly performing user as if all of the user's relationship links were to users of feedback rating.

$$\phi_{pess}(i) = \begin{cases} 0 & \text{if } L_{[k]}(i) < 1 - \delta \\ 1 & \text{otherwise} \end{cases}$$

A pessimistic approach may be appropriate in circumstances when relationships with malicious users are highly correlated (as in a malicious clique) or when malicious users in the network are considered extremely dangerous. In this second case, even a single relationship link to such a dangerous user would warrant a severe correction to the link quality of the recommending user.

Hop-Based Correction: In contrast, the hop-based correction factor seeks to provide a balance between the optimistic and pessimistic correction factors by considering the number and the length of the paths emanating from a user that reach bad users. A *path* in the social network from user i to user j is a sequence of users: $path(i, j) = \langle x_0, x_1, ..., x_n \rangle$ (where $i = x_0$ and $q = x_n$) such that there exists a relationship link between successive nodes in the path, $x_{l+1} \in rel(l)$, for $0 \leq l \leq n - 1$. We say a path reaches a bad user if the feedback rating for the user is less than some threshold δ. We call such a path a *bad path*.

For a bad path of length l originating at user i, we associate a hop-based correction factor $\phi_{hop,l}(i)$, where $0 \leq \phi_{hop,l}(i) \leq 1$. By default, we let $\phi_{hop,l}(i) = 1$ if there are no bad paths of length l originating from i. The hop-based discount factor can then be calculated as the product of the constituent discount factors: $\phi_{hop}(i) = \prod_{l=1}^{k} \phi_{hop,l}(i)$.

Selecting the appropriate hop-based correction factor is important, and there are a number of possible approaches. In this paper, we advocate an exponentially decaying correction factor. Beginning with a user-defined factor ψ ($0 < \psi < 1$) to set the initial hop-based correction for bad paths of length 1, i.e., $\phi_{hop,1}(i) = \psi$, the exponential approach tunes this correction closer to 1 as the bad path length increases:

$$\phi_{hop,l}(i) = 1 - (1 - \psi)\psi^{l-1}$$

meaning that longer bad paths result in a less severe correction to a user's link quality than do shorter paths. Starting with an initial correction factor ψ close to 0 will result in a more pessimistic correction, whereas ψ close to 1 is intuitively more optimistic.

5. EVALUATION

In this section, we evaluate the SocialTrust framework through simulations of community-based information sharing over real social network data. We focus on three aspects: (i) a comparison of SocialTrust versus alternative trust models; (ii) the study of link quality; and (iii) an evaluation of SocialTrust in the presence of strategies attempting to subvert its effectiveness, including clique formation and collusive feedback. We find that the SocialTrust framework supports robust and tamper-resilient trust ratings even when large portions of the social network engage in behavior intended to undermine its effectiveness.

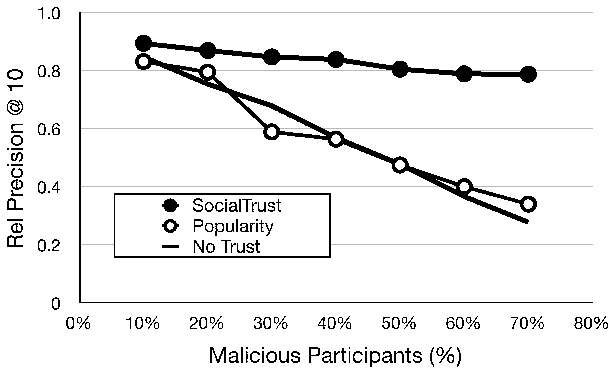

Figure 1: SocialTrust vs. Basic Trust Models

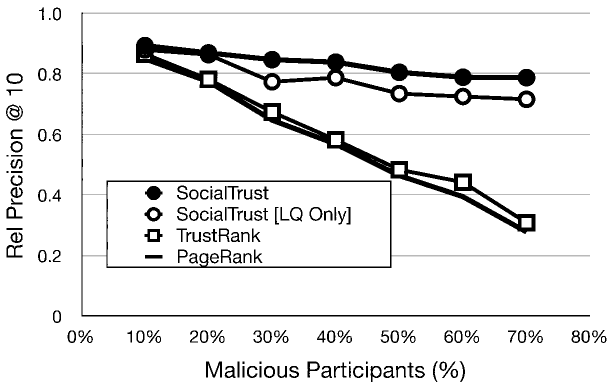

Figure 2: SocialTrust vs. PageRank and TrustRank

5.1 Experimental Setup

Data: All of the experiments reported in this paper rely on data collected from MySpace, the largest social networking site and one of the few that provides open access to public user profiles. We ran multiple parallel crawlers over MySpace in July 2006, beginning from a random sample of seed profiles. The crawlers followed the relationship links listed on each profile's front page in a breadth-first traversal of MySpace, resulting in a collection of 891,197 full-text profiles. Based on these profiles, we generated a directed graph consisting of 5,199,886 nodes representing both the collected full-text profiles and additional referenced profiles and 19,145,842 relationship links. A more detailed study of this dataset can be found in [9].

Application Scenario: As an application setting for evaluating the quality of SocialTrust, we consider a scenario in which an *originating user* has an information need (e.g., looking for a job in Texas, finding a good restaurant) for which she can use her social network. The basic scenario is this: a user browses her relationships up to some radius looking for candidate users to ask; based on an analysis of their profiles, she constructs a set of candidate users who might satisfy her information need; based on the provided trust ratings, she selects the top-k most trusted candidate users; she asks all top-k; if she is satisfied, she provides positive feedback to the trust manager; otherwise, she provides negative feedback.

Simulation Setup: The simulation begins from a cold start, in which each user in the network is assigned a default trust score. Thereafter, users are randomly selected to begin a browsing session for a particular information need, they report their feedback to the trust manager, and at regular intervals the trust manager calculates the trust score for each user in the network for use in the next cycle. For each browsing session, we simulate a large browse over a relationship-based trust group with radius 7, in which the originating user browses using random selection with up to eight random neighbors selected at each step. We intentionally select such a large trust group (covering on average 26k users or 0.5% of the network) to stress-test the quality of the trust values since more malicious users will be available to corrupt the quality of responses in each browsing session.We model an originating user's information need using a simple unigram information retrieval model: a "query" term is randomly selected from the space of all MySpace profiles, weighted by the number of profiles in which it occurs. A profile encountered during browsing is considered a candidate based on a simple binary match between the selected term and the user's profile.

User Behavior: We model two types of users: (i) malicious users, who always provide an irrelevant response when asked; and (ii) legitimate users, who sometimes accidentally provide an irrelevant response when asked.

Evaluation Metric: For a query q, let R^+ denote the set of relevant users for q throughout the entire space of users and let R_n denote the n top-ranked candidate users (by trust value). We measure a focused version of the standard precision measure that considers the quality of the responses in the top-n (the relative precision @ n): $prec_n = \frac{|R^+ \cap R_n|}{min(|R_n|, n)}$. This relative precision metric measures the effectiveness of trust ratings by considering the quality of the top responses for a user's information need, even if fewer than n are returned. The traditional precision and recall measures provide little distinguishing power since malicious users may overwhelm an originating user with many poor quality responses. We measure the average performance over many browsing sessions starting from many different originating users, so we can identify system-wide quality metrics for comparing trust models.

Trust Calculation: All trust calculations are performed using the Jacobi method for 25 iterations and a mixing parameter $\lambda = 0.85$. In all of our experiments, a simulation cycle consists of 5,000 browsing sessions. There are 30 simulation cycles in total. For each query, users provide feedback over the top-20 most trusted users they encounter. We report results over the last 5,000 browsing sessions, averaged over five simulation runs. In all of the reported experiments, we use the SocialTrust trust model described in Equation 5. For the link quality component, we rely on the scoped random walk model with scope of $k = 3$ and an exponential correction factor with $\psi = 0.5$ and $\delta = 0.5$. We shall revisit some of these assumptions in the following experiments.

5.2 Comparing Trust Models

We first evaluate the quality of SocialTrust against alternative trust models and for varying degrees of user manipulation within the network. For each of the trust models, we consider seven scenarios: 10% of the network is mali-

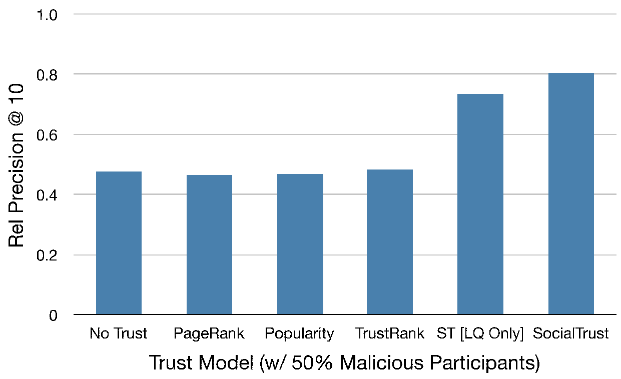

Figure 3: Detail: Comparing Trust Models

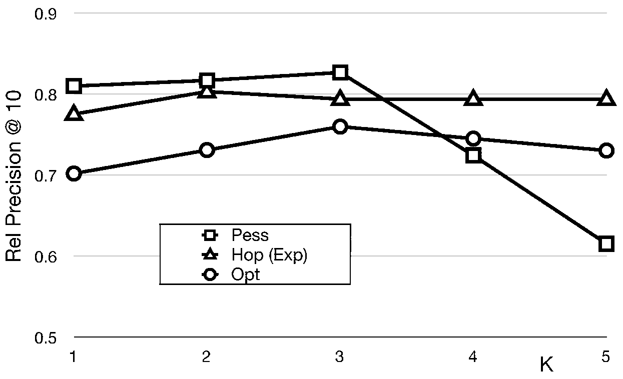

Figure 4: Evaluating Link Quality

cious, 20%, ..., 70%. When asked, malicious users provide a corrupt (irrelevant) response with 100% probability; other users respond with a corrupt result with 5% probability. In all cases, if 100% of the network is malicious, the trust ratings are meaningless and the overall precision drops to 0.

In Figure 1, we compare SocialTrust against the *No Trust* case – in which a user randomly selects among the users encountered in the browsing session – and a simple trust model based on the *Popularity* of the user in the network based on the number of relationships she participates in: $Tr_{q,pop}(i) = |rel(i)|$. In both cases, we see that the relative precision for SocialTrust is resilient to the increase in malicious users, whereas the *No Trust* and *Popularity* models degrade severely. With an increasing number of malicious users in the network, neither the *No Trust* model nor the *Popularity* model gives the unsuspecting user any assurances as to the quality of the users in the network.

Given that SocialTrust outperforms these naive models, how well does it perform against more sophisticated ones? In Figure 2, we compare SocialTrust to several related trust models adapted from the Web and P2P domain to online social networks. We consider a PageRank-based trust model that considers only the relationship structure of the social network; a TrustRank-based model that uses feedback ratings as a priori trust (which is equivalent to EigenTrust from the P2P domain); the preliminary SocialTrust [LQ Only] model that incorporates relationship link quality only but no feedback ratings (which is similar in spirit to credibility-based link analysis explored in the Web domain in [8]); and the final SocialTrust model.

First, both the *PageRank* and *TrustRank* models degrade severely, performing nearly as poorly as the naive *Popularity* and *No Trust* approaches. At first glance, the fall in precision for these models may be surprising, but consider that malicious users are distributed throughout the network, meaning some of the initially most trusted users are malicious. When a proportion of these highly-trusted users behave maliciously, PageRank and TrustRank have no mechanism for correcting this bad behavior. In contrast, the SocialTrust model incorporates link quality and feedback ratings into the trust assessment so that bad behavior is punished, and so the resulting precision measures are resilient to the presence of a large fraction of malicious users in the network. This is especially encouraging since the feedback ratings available in one simulation round may be incomplete for users who have not yet been rated in previous rounds. Also note that the inclusion of relationship link quality (SocialTrust [LQ Only]) provides the single biggest improvement in precision, since it reduces the influence of users who engage in poor quality relationships. When coupled together, both feedback ratings and link quality provide the best performance (SocialTrust).

To further illustrate, we compare all of the trust models in Figure 3 for the scenario when 50% of the network is malicious. Here, we can see the importance of considering relationship link quality (in the difference between SocialTrust [LQ Only] and the other random models), as well as the important but less significant impact of incorporating feedback ratings (in the difference between SocialTrust [LQ Only] and SocialTrust).

5.3 Impact of Relationship Link Quality

Since the relationship link quality is such an important factor, we next compare several versions. We additionally consider the optimistic approach for $k = 1$ to $k = 5$, the pessimistic approach for $k = 1$ to $k = 5$ (with $\delta = 0.5$), as well as the exponential hop-based approach for $k = 1$ to $k = 5$. In Figure 4, we report the relative precision @ 10 for the scenario when 50% of the users in the network are malicious, but with the different approaches for computing relationship link quality incorporated into the trust model.

First, the optimistic and hop-based approach are stable and perform fairly well as the scope parameter k increases. These approaches penalize a candidate user's relationship link quality score in proportion to the distance of malicious users from the candidate user. Direct links to malicious users result in a lower link quality score than paths of multiple hops to malicious users. In contrast, the pessimistic approach results in a worsening of precision as the scope increases. As k increases, most users have at least one path to a malicious user and are assigned a 0 or low relationship link quality score. As the link quality score approaches 0 for nearly all users in the network, the rankings induced from the trust model become random, and so we see the precision fall considerably.

5.4 Clique Formation

In our previous experiments, malicious nodes enter the network randomly. Suppose instead that malicious nodes seek to form cliques in the social network so that they can leverage their tightly-coupled relationship structure to over-

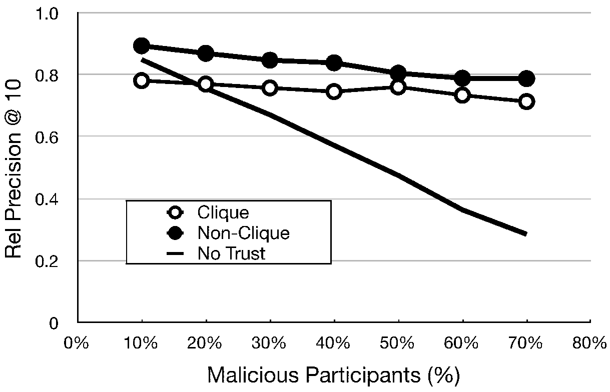

Figure 5: Effectiveness of Clique Strategies

power SocialTrust. Rather than randomly assigning users to be malicious, we now construct malicious cliques. The setup works like this: first a node is randomly selected and assigned to be a malicious node, then up to three-hops of its neighbors are also assigned to be malicious. We repeat this process until 10% of the network is malicious. This overall procedure continues for the 20% case, 30% case, up to the 70% case.

In Figure 5 we report the relative precision @ 10 for SocialTrust over this clique-based strategy (*Clique*). As points of comparison, we also show the performance of SocialTrust over the original non-clique strategy (*Non-clique*), as well as the performance of the *No Trust* strategy over the clique-based strategy. Even in the presence of cliques, the SocialTrust approach provides resilient rankings as the fraction of malicious users increases. We attribute the success of the SocialTrust approach to its incorporation of relationship link quality, so that the influence of malicious cliques over the aggregated trust ratings is reduced.

5.5 Subverting Feedback Ratings

Suppose that in addition to providing irrelevant answers when asked, that malicious users also attempt to subvert the feedback ratings. So far, we have used the Trust-Aware Restricted Voting at the end of each simulation cycle, where a user's feedback is proportional to his trust rating. In this final experiment, we consider the other two voting schemes discussed in Section 4.4 – open voting and restricted voting.

Recall that the Trust-Aware approach allots a voting share to each user based on his trust value, so that more trusted users have greater sway over the feedback ratings of other users than do lowly trusted users. For the restricted voting case, each user is allotted an equal voting share for distributing among the users in his trust group who have answered its queries in the past. In the open voting case, there are no constraints on the number of votes cast by any user.

For each voting scheme, we assume that a malicious user always provides negative feedback for legitimate users, regardless of the quality of the answer provided; a legitimate user provides honest feedback. For the open voting case, we assume that malicious users ballot stuff the voting process, resulting in feedback ratings for legitimate users randomly distributed between $[0, 0.1]$. Malicious users receive high feedback ratings randomly distributed between $[0.9, 1]$.

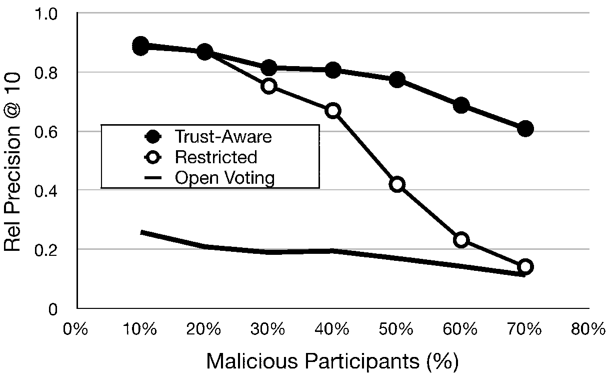

Figure 6: Comparing Feedback Schemes

In Figure 6, we compare the performance of the SocialTrust framework over each voting scheme. As the network tips over 50% malicious, the restricted voting case begins a steep decline. In this scenario, there are more malicious users in the network and so (regardless of their past trust values), they can promote other malicious users, so that in the following round these malicious users receive a boost in feedback rating (and hence, link quality, and ultimately, trust). For the open voting scheme, we see that precision is very low across the scenarios. Even a small percentage of malicious nodes can subvert the feedback ratings of legitimate users (and promote the scores of other malicious users), so that the derived trust ratings favor malicious users.

In contrast, the trust-aware voting scheme is fairly resilient; as more and more malicious users enter the network, the highly-trusted users manage to keep them under control. The robustness of the SocialTrust model, even with large portions of the network providing dishonest feedback, can be partially attributed to our model of how malicious users enter the network. In our simulations, malicious user are activated in 10% chunks. Since trust and feedback ratings are linked from round-to-round, the votes of legitimate users in one round can deter the malicious users from receiving high trust scores in the following round. In contrast, if 70% of the entire network were to suddenly behave maliciously, we would observe a steep degradation in precision. Based on this observation, we are studying additional feedback countermeasures to incorporate into future revisions of SocialTrust.

6. CONCLUSION

In this paper, we have presented the design and evaluation of the SocialTrust framework for aggregating trust in online social networks and provided the first large-scale trust evaluation over real social network data. The proposed framework supports tamper-resilient trust establishment in the presence of large-scale manipulation by malicious users, clique formation, and dishonest feedback. We have seen how trust group feedback and distinguishing between relationship quality and trust can result in more resilient trust ratings than in algorithms like PageRank and TrustRank.

In our future work, we are interested in developing context-aware extensions of SocialTrust so that the network may support multiple trust views of each user depending on the context. We also see opportunities to augment the evaluation of relationship link quality, so that it considers more sophisticated features like the nature, duration, and value of each relationship. On the implementation side, we continue work on a SocialTrust-powered community platform that can be layered on top of existing social networks.

7. ACKNOWLEDGMENTS

This work is partially supported by faculty startup funds from Texas A&M University and the Texas Engineering Experiment Station and by grants from NSF CyberTrust, NSF CSR, NSF ITR, an IBM faculty award, IBM SUR grant, and an AFOSR grant. We also thank the anonymous reviewers for providing many valuable comments.

8. REFERENCES

[1] K. Aberer and Z. Despotovic. Managing trust in a Peer-2-Peer information system. In *CIKM*, 2001.

[2] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187–203, July 2005.

[3] L. Backstrom, D. P. Huttenlocher, J. M. Kleinberg, and X. Lan. Group formation in large social networks. In *KDD*, 2006.

[4] S. Bao, G. Xue, X. Wu, Y. Yu, B. Fei, and Z. Su. Optimizing web search using social annotations. In *WWW*, 2007.

[5] E. Bertino, E. Ferrari, and A. C. Squicciarini. Trust-x: A peer-to-peer framework for trust establishment. *IEEE Trans. Knowl. Data Eng.*, 16(7):827–842, 2004.

[6] C. Boyd. Teenagers used to push Zango on MySpace. `http://www.vitalsecurity.org`, 2006.

[7] C. R. Brooks and N. Montanez. Improved annotation of the blogosphere via autotagging and hierarchical clustering. In *WWW*, 2006.

[8] J. Caverlee and L. Liu. Countering web spam with credibility-based link analysis. In *PODC*, 2007.

[9] J. Caverlee and S. Webb. A large-scale study of MySpace: Observations and implications for online social networks. In *2nd International Conference on Weblogs and Social Media (AAAI)*, 2008.

[10] F. Cornelli, E. Damiani, and S. D. Capitani. Choosing reputable servents in a P2P network. In *WWW*, 2002.

[11] A. Doan, R. Ramakrishnan, F. Chen, P. DeRose, Y. Lee, R. McCann, M. Sayyadian, and W. Shen. Community information management. *IEEE Data Engineering Bulletin*, March 2006.

[12] P. S. Dodds, R. Muhamad, and D. J. Watts. An experimental study of search in global social networks. *Science*, 301(5634):827–829, August 2003.

[13] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of trust and distrust. In *WWW*, 2004.

[14] Z. Gyöngyi, H. Garcia-Molina, and J. Pedersen. Combating web spam with TrustRank. In *VLDB*, 2004.

[15] T. N. Jagatic, N. A. Johnson, M. Jakobsson, and F. Menczer. Social phishing. *CACM*, 50(10):94–100, 2007.

[16] S. Kamvar, B. Yang, and H. Garcia-Molina. Secure score management for peer-to-peer systems. Technical report, Stanford University, 2004.

[17] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The EigenTrust algorithm for reputation management in P2P networks. In *WWW*, 2003.

[18] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina. Combating spam in tagging systems. In *AIRWeb '07: Proceedings of the 3rd international workshop on Adversarial information retrieval on the web*, 2007.

[19] R. Kumar, J. Novak, and A. Tomkins. Structure and evolution of online social networks. In *KDD*, 2006.

[20] R. Li, S. Bao, Y. Yu, B. Fei, and Z. Su. Towards effective browsing of large scale social annotations. In *WWW*, 2007.

[21] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. Geographic routing in social networks. *Proceedings of the National Academy of Sciences*, 102(33):11623–1162, 2005.

[22] S. Marti and H. Garcia-Molina. Taxonomy of trust. *Computer Networks*, 50(4):472–484, March 2006.

[23] P. Massa and P. Avesani. Controversial users demand local trust metrics. In *AAAI*, 2005.

[24] S. Milgram. The small-world problem. *Psychology Today*, pages 60 – 67, May 1967.

[25] R. Monastersky. The number that's devouring science. *The Chronicle of Higher Education*, October 2005.

[26] H. Ozbay. *Introduction to feedback control theory*. CRC Press Inc, 1999.

[27] L. Page et al. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford University, 1998.

[28] M. Richardson, R. Agrawal, and P. Domingos. Trust management for the semantic web. In *ISWC*, 2003.

[29] S. Rubin, M. Christodorescu, V. Ganapathy, J. T. Giffin, L. Kruger, and H. Wang. An auctioning reputation system based on anomaly detection. In *CCS*, 2005.

[30] M. Sanchez. Pranksters posting fake profiles on MySpace. *http://www.dfw.com/*, 2006.

[31] M. Srivatsa, L. Xiong, and L. Liu. TrustGuard: Countering vulnerabilities in reputation management for decentralized overlay networks. In *WWW*, 2005.

[32] S. Wasserman and K. Faust. *Social network analysis*. Cambridge University Press, Cambridge, 1994.

[33] D. J. Watts. Networks, dynamics, and the small world phenomenon. *American Journal of Sociology*, 105(2):493–527, 1999.

[34] B. Wu, V. Goel, and B. Davison. Topical TrustRank: Using topicality to combat web spam. In *WWW*, 2006.

[35] L. Xiong and L. Liu. Supporting reputation-based trust for P2P electronic communities. *TKDE*, 16(7), 2004.

[36] S. A. Yahia, M. Benedikt, and P. Bohannon. Challenges in searching online communities. In *IEEE Data Engineering Bulletin*, 2007.

[37] B. Yu and M. P. Singh. A social mechanism of reputation management in electronic communities. In *Cooperative Information Agents*, 2000.

[38] J. Zhang, M. S. Ackerman, and L. Adamic. Expertise networks in online communities: Structure and algorithms. In *WWW*, 2007.