Keeping Peers Honest in EigenTrust

Zoë Abrams[*], **Robert McGrew**[†], and **Serge Plotkin**
Stanford University
{zoea, bmcgrew, plotkin}@stanford.edu

Abstract

A fundamental consideration in designing successful trust scores in a peer-to-peer system is the self-interest of individual peers. We propose a strategyproof partition mechanism that provides incentives for peers to share files, is non-manipulable by selfish interests, and approximates trust scores based on EigenTrust[5]. The basic idea behind the partition mechanism is that the peers are partitioned into peer groups and incentives are structured so that a peer only downloads from peers in one particular peer group. When peers are myopic, we show that the *total* error in the trust values decreases exponentially with the number of peer groups, but when peers plan for the future, the total error decreases linearly with the number of peer groups. There is a direct tradeoff here between approximating EigenTrust well and keeping the trust values we are approximating useful.

Introduction

Peer-to-peer networks have been a focus of much interest in the distributed systems community, due to their advantages in scalability and robustness over traditional client-server architectures. However, the openness and symmetry that gives P2P systems their advantages results in their vulnerability to attacks and manipulations, as well as to free-riders. Deployed P2P file-sharing networks today have problems with malicious peers who share inauthentic files and free-riding peers who download files but do not share them.

There is a considerable amount of recent research that focuses on the design and development of systems which rate peers on their likelihood of giving authentic files [5][2][1]. In these works, it is assumed that there are two types of peers. On the one hand, there are honest peers who, though they might free-ride unless rewarded for sharing, will never share inauthentic files or cheat the trust system. On the other hand, there are malicious peers who attempt to minimize the number of successful downloads in the network by any possible means. These reputation systems succeed at isolating malicious peers from the network and encouraging honest peers to share files by rewarding them with high trust values and with a better quality of service.

Unfortunately, honest peers, who had no desire to manipulate the system before the introduction of trust, now have an incentive to lie to the system in order to improve their trust. A selfish peer will naturally submit the recommendation, true or false, that will maximize its trust and therefore its quality of service. Unlike malicious peers, these selfish peers do not wish to disrupt the network but merely to maximize their own utility. Nonetheless, if each peer provides false recommendations, the trust system as a whole will be unable to discriminate between honest and malicious peers. The gaming of real-world reputation systems such as Amazon [11] and Kazaa demonstrate that this concern is not merely theoretical.

We address this problem by creating a trust system in which a peer's recommendation does not affect its own score. Thus, the peer will have no incentive to manipulate its recommendation to increase its trust. We will proceed by approximating EigenTrust [5], an existing trust system similar to PageRank [12], with a system in which no peer can unilaterally manipulate its trust score.

In our trust system, time is divided into rounds, with downloads in one round determining trust in the next. A trust score is *myopically non-manipulable* if a peer cannot submit falsely to improve its trust score for the next round; it is *strongly non-manipulable* if he cannot submit falsely to improve its score in any future round.

Before we discuss our positive results, we note a negative result: if, given some joint set of legal downloads by all peers, some peer can gain an amount 2δ by altering its report of its downloads, then there exists no myopically non-manipulable trust score that approximates EigenTrust with error less than δ. We must thus restrict the possible joint sets of downloads so that a non-manipulable trust score which closely approximates EigenTrust exists. In our results, we restrict the query topology by dividing peers into colors arranged in a directed cycle. We only allow peers to query and download from their successor color.

Our positive results are as follows:

1. We show a myopically non-manipulable trust score that approximates EigenTrust with error decreasing exponentially in the number of colors.

2. We show a strongly non-manipulable trust score that approximates EigenTrust, but with error that decreases only linearly in the number of colors.

[*]Research supported by NSF/CCR 0113217-001.
[†]Research supported in part by NSF under ITR IIS-0205633.

Related Work

The self-interest of individual peers has been recognized in previous work as a fundamental consideration in designing successful P2P systems. Dutta et al. [8] discuss issues of self-interest in the design of trust rating schemes, including the free-rider problem and the challenge of designing collusion-proof systems. In [3], Ng et al. outline guiding principles for a vision of strategyproof computing in P2P systems. Buragohain et al. [7] study the interaction of strategic and rational peers from a game theoretic perspective, and propose a differential service-based incentive scheme to encourage file-sharing participation. Finally, in [9], a simple Selfish Link-based Incentive mechanism (SLIC) is presented for P2P file sharing systems, where the amount of service a peer receives depends on the amount of service it provides.

Model

In this section, we outline our interaction model, the strategic model of the agents, and the requirements that a non-manipulable trust system must satisfy.

As in EigenTrust, our model is divided into rounds in which peers interact by making queries and downloading files. At the end of each round, the record of authentic and inauthentic downloads is used to calculate the trust values for the next round. In order to simplify the model, we assume the existence of a non-strategic third-party known as the *center* which is available to calculate peers' trust. In practice, our mechanism is capable of operating in a distributed environment using either the algorithm in [6] or the efficient algorithm which we will describe in the Distributed Implementation section.

Strategic Model

Instead of merely considering honest and malicious peers, we assume that the objective of each peer is to selfishly maximize its utility function. In this work, we assume that the utility of a selfish peer is its trust score. We distinguish between two types of agents: *myopic* agents who only consider their utility in the next round, and *strongly rational* agents who we assume for concreteness discount their future utilities with some discount factor γ. The extension to other means of weighing utilities over time is straightforward.

We model peers as trust-score maximizers because EigenTrust rewards peers for having a high trust score in order to incent peers to share. While we do not fully generalize this argument here, any well-designed system that prevents free-riders should reward peers for increasing the number of files they upload to other peers in the network, and the primary means for a peer to increase the number of files uploaded to other peers is by increasing its own trust score.

We define the action space of a peer i in round r as the set of all possible downloads $\hat{d}_i^r \in \mathcal{D}$ that i may report to the center. Although there are many strategic behaviors which a peer could adopt that do not depend on the trust score system, such as misreporting its trust score to querying nodes or cheating during the distributed computation of the trust scores, these forms of cheating can be prevented or mitigated using previously studied techniques [5][1], and the methods described in these papers can be used in conjunction with the mechanism we propose. Therefore, as we are primarily interested in creating a non-manipulable trust score system, rather than in creating a secure computing environment, we do not consider other forms of manipulation in the strategy space of selfish peers.

Formally, at the beginning of each round r, each peer i makes a set of queries Q_i^r. From these queries, the peer receives a set of authentic and inauthentic downloads, which we denote by d_i^r. The entire set of downloads received by all peers we denote as d^r, while we refer to the set of downloads received by all agents other than i as d_{-i}^r. We refer to the set of all possible downloads as $\mathcal{D}$. At the end of each round, each peer submits its report $\widehat{d_i^r}$ of the downloads received. These reports are used to calculate the trust t_i^{r+1} for each peer in the next round according to the trust function $T_i(\hat{d}_i, \hat{d}_{-i})$. We sometimes omit the superscript r in our notation when the round is clear from context.

Solution Concepts

Although each peer knows the set of downloads which it made during the round, this information is unknown the to the mechanism designer. We cannot use the standard techniques of mechanism design to induce revelation of this information, however, because this private information does not affect the peer's preferences over outcomes. This fact also implies that the strategy space of an agent is not, as is standard, a mapping from its private information to its report, because a peer's utility from its report is independent of its private information. Instead, each action is simply a declaration. Furthermore, although we will not prove this statement here, if we attempt to obtain this information from the other peers, we will only induce peers to make unnecessary downloads for the sake of increasing their trust values. We must therefore require that each possible report of each peer is as good for that peer as any other report; thus no peer has any incentive to lie.

Formally, we define the single-round game $G(\mathcal{D}, T)$ as $\langle N, \mathcal{D}, T \rangle$, the game with players N, actions of i as $\hat{d}_i \in \mathcal{D}_i$, and payoff for i as $T_i(\hat{d}_i, \hat{d}_{-i})$. A *dominant strategy equilibrium* (DSE) of this game is a profile $\hat{d}^* \in \mathcal{D}$ of actions such that for every player $i \in N$ we have $T(\hat{d}_i^*, \hat{d}_{-i}) \geq T(\hat{d}_i, \hat{d}_{-i})$ for all $\hat{d} \in \mathcal{D}$. In other words, $\hat{d}_i^*$ is a best response to any strategy $\hat{d}_{-i}$ of the other players. We wish to design a reputation system in which peers will honestly report the downloads they received, no matter what those were and no matter what other peers choose to report; that is, for every $\hat{d} \in \mathcal{D}$, $\hat{d}$ is a DSE. As discussed above, this is equivalent to making each peer indifferent between its actions $\hat{d}$.

We define the game $H(\mathcal{D}, T)$ as the multi-round game in which each player i has payoff $\sum_r \gamma^r T_i(\hat{d}^r)$. A DSE in this multi-round game is defined analogously to the single-round game above, except considering extended strategies over the games rather than single-round actions. If multi-round-game strategy σ is a DSE of H, σ_i must be a best response to any arbitrary reporting strategy of the other peers. For instance, σ_i must be a best response to the following strategy: i rec-

ommends a particular peer k in round r, each peer $j \neq i$ will recommend i in every round following r. Clearly, the only best response to this strategy is for i to recommend k in round r. To avoid these situations, we make the natural restriction that each peer j's strategy for choosing $\hat{d}_j^r$ be dependent only on the history of actions of peers who can affect j's trust score. This restriction allows the possibility of Tit-for-Tat-style collusive strategies between peers but disregards the possibility of collusion in which one party has nothing to gain.

If we can assume that our peers have a myopic utility function, we need only require that any joint report of downloads be a DSE in the single-round game G; if, on the other hand, the peers are strongly rational, we must require that honest reporting be a DSE in the multi-round game H.

Formally, we define two notions of non-manipulability:

1. Myopic Non-Manipulability: For every $d \in \mathcal{D}$, $\hat{d} = d$ is a DSE of the single-round game $G(\mathcal{D}, T)$. This is equivalent to the condition that $\forall \widehat{d} \in \mathcal{D} : T_i^{r+1}(\widehat{d}_i^r, \widehat{d}_{-i}^r)$ is independent of $\widehat{d}_i^r$. This corresponds to a myopic player being indifferent between its recommendations at the current round.

2. Strong Non-Manipulability: For every $(d^1, d^2, ...) \in \mathcal{D}^*$, $(\hat{d}^1, \hat{d}^2, ...) = (d^1, d^2, ...)$ is a DSE of the multi-round game $H(\mathcal{D}, T)$. This is equivalent to the condition that $\forall \sigma \in \Sigma, \forall r' > r : T_i^{r'}(\sigma^{r'})$ is independent of σ_i^r; that is, each peer's trust is independent of all of its previous recommendations. This corresponds to a strongly rational player i being indifferent between its recommendations at r, given any joint future strategy of itself and the other players.

Clearly, strong non-manipulability implies myopic non-manipulability, but the converse does not hold.

We note that, although peers have no incentive to lie, they also have no incentive to tell the truth. However, since a peer's utility is independent of its private information, it is not possible to make a peer have a strict preference for truthtelling. Moreover, we note that players already discover whether their files are authentic or inauthentic as a normal part of the download process and it is a simple matter to report this to their implementation of the protocol. Since the default implementation of the protocol reports this information truthfully, there is no reason for anyone to program a lying version of this protocol. In practice, we would expect to see truthtelling behavior in any system with these non-manipulability properties.

Of course, a non-manipulable trust score system would be of no use if it did not alienate malicious peers. Our approach will be to show that we do not give more than a certain amount of additional trust to the set of malicious peers above and beyond what EigenTrust gives. If EigenTrust alienates malicious peers and our system closely approximates the trust scores given by EigenTrust, we will be satisfied that our system alienates malicious peers.

The EigenTrust Algorithm

The EigenTrust algorithm is intended to compute a trust score for agents which indicates how likely a peer is to be malicious. We choose EigenTrust to approximate because it has been shown in simulations to be successful at alienating malicious peers and also because theoretical analysis of PageRank[4] can be applied to EigenTrust to show a bound on manipulability

As we will present a modified version of EigenTrust in our work, we will briefly describe the EigenTrust trust score.

For some $q \in Q_i$, $j \in serveri(q)$ signifies that j responded affirmatively to i's query. If in a round r a peer i has had $sat(i,j)$ satisfactory downloads from j and $unsat(i,j)$ unsatisfactory downloads, let $s_{ij} = max(sat(i,j) - unsat(i,j), 0)$ and $d_{ij} = \frac{s_{ij}}{\sum_k s_{ik}}$. In words, d_{ij} is a normalized measure of how much i trusts j. EigenTrust assumes the existence of a distribution p over *pre-trusted peers* which is commonly known in the system. Define $\widehat{D}$ as the matrix $[\widehat{d}_{ij}]$ and P as the matrix $[p_{ij} = p_j]$. We define a probability ϵ, known as the *teleport probability* for historical reasons, which measures how much trust the pre-trusted peers receive due to their pre-trusted status. In practice, ϵ is usually set to 0.2.

The EigenTrust value of each node i is computed as i's share of the stationary distribution of a Markov chain with transition matrix $M = (1-\epsilon)\widehat{D}^T + \epsilon P$. So $t_i = T_i(\hat{d})$ can be calculated as the principal right eigenvector of M.

We now define the EigenTrust algorithm:

Initialization Initialize the trust uniformly, assigning every peer i trust $t_i^0 = \frac{1}{n}$.
Run Transactions Until the end of the current round r, each peer i successively makes its queries $q \in Q_i^r$. From the peers $server_i(q)$, a single peer j is selected to serve the file with probability $t_j^r / \sum_{k \in server_i(q)} t_k^r$.
Compute Trust Values At the end of round r, each peer i sends the entire report $\widehat{d}_i^r$ of all its downloads d_{ij}^r to the center. $t_i^{r+1} = T_i(\hat{d}_i^r) = eig_i((1-\epsilon)\widehat{D^r}^T + \epsilon P)$

The EigenTrust Algorithm has the attractive property that it is *upload maximizing*: a peer's decision to share an authentic file *always* results in an increase in that peer's trust value, and therefore a selfish peer will want to maximize the number of uploads it performs.

Approximate Trust and δ-Equilibria

As shown in [5], EigenTrust is effective in simulations at alienating malicious peers from the network. In addition, the matrix D can be viewed as the link matrix for a graph in which one's downloads correspond to outgoing links in the graph, and EigenTrust is exactly the application of PageRank to that graph. Therefore, we can apply analysis of the susceptibility of PageRank to perturbations [4] to give theoretical arguments for the effectiveness of EigenTrust. By a trivial extension of this work, we can show that a selfish peer cannot increase its trust by more than a factor of $\frac{(1+\epsilon)}{\epsilon}$

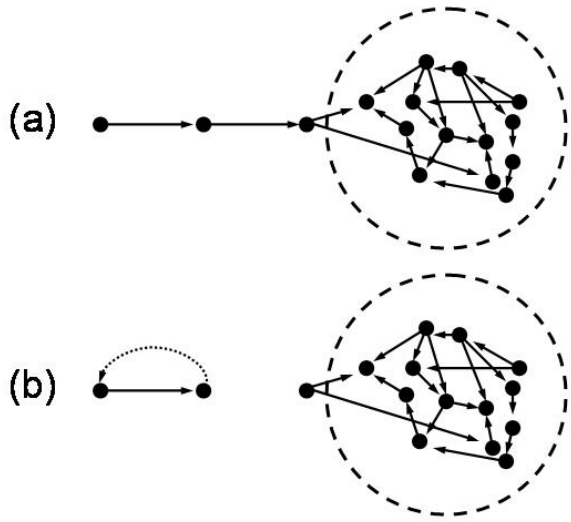

Figure 1: **Manipulation Example**
Part (a) shows the actual download graph; in part (b), the dotted line is a manipulation by the middle node to increase its trust score.

by manipulating its report to the center. Since a typical setting for ϵ in real systems is 0.2, this implies that no peer can increase its trust by more than a factor of 6. If a malicious peer's true trust score is already low, even this worst-case manipulation will not give it enough additional trust to disrupt the network.

While a manipulation to gain a factor of 6 may not disrupt the network, it is certainly enough to incent a selfish peer to lie about its recommendation. To maximize its trust, a peer ought always to recommend a peer that recommended it. Consider the download graph of Figure 1, and assume a uniform distribution for pre-trusted peers over all n peers: if the middle node reports a download from the right node as in Figure 1(a), it will have trust $(2-\epsilon)\epsilon/n$. If, on the other hand, it reports a download from the left node as in Figure 1(b), it will have trust $1/n$. Scenarios of this sort with a node that has only one incoming link may arise often in real systems and the ratio of increase is independent of the number of peers.

As we will see, some classes of download graphs such as acyclic graphs or graphs without tight loops prevent any peer from gaining much trust by unilaterally manipulating its outgoing links. We can define the notion of an δ-equilibrium: an action profile $\hat{d}^* \in \mathcal{D}$ is a δ-equilibrium of $G(\mathcal{D}, T)$ if for every player $i \in N$ we have $T(\hat{d}_i, \hat{d}^*_{-i}) \leq T(\hat{d}^*_i, \hat{d}^*_{-i}) + \delta$ for all $\hat{d}_i \in \mathcal{D}$.

We wish to approximate our chosen trust score T with a non-manipulable trust score $\tilde{T}$. Ideally, we would find a non-manipulable $\tilde{T}$ which approximates T on every download graph d. In fact, the existence of a non-manipulable trust score that approximates EigenTrust over a set of download graphs $\mathcal{D}$ requires every action profile in $\mathcal{D}$ to be a δ-equilibrium.

Theorem 1 *If some action profile d in the single-round-game $G(\mathcal{D}, T)$ is not a 2δ-equilibrium, then there does not exist a myopically non-manipulable trust score $\tilde{T}$ s.t. $\sum_{i \in N} |T_i(d) - \tilde{T}_i(d)| \leq \delta$.*

The proof is given in Appendix A.

In fact, we can find an infinite family of download graphs in which a peer can increase its trust by an additive constant slightly less than 1/2. Thus, if we wish to find even myopically non-manipulable trust score with small error, we must restrict ourselves to classes of download graphs in which players can manipulate their trust scores only to a small extent.

Cyclic Partitioning

We now consider modifications to the basic EigenTrust algorithm such that our new trust score is myopically non-manipulable. By the results of the previous section, we note that, if a class of download graphs has a bad maximum manipulability, any non-manipulable trust score will have high error on some graph. However, since we are designing the P2P system, we can restrict the topology of the network in a way that only allows download graphs with low manipulability. This restriction is natural in that the upload maximizing property *only* holds for the downloads the designer desires, and thus there are no incentives for peers to share files with peers other than those chosen by the designer.

Algorithm

We make changes to the EigenTrust algorithm as described below, allowing peers to enter and leave at the end of each round.

Initialization At the initialization of our algorithm, the center partitions the peers evenly into colors, where $C = \{c_1, c_2, ...c_m\}$ is the set of partitions. Each color has either $\lfloor \frac{n}{m} \rfloor$ or $\lceil \frac{n}{m} \rceil$ peers. The center arranges the colors into a directed cycle chosen uniformly at random. $\forall c \in C$, let $pred(c)$ be the color which is the predecessor of c in the cycle and $succ(c)$ the successor of c. We restrict the distribution p over pre-trusted peers to assign an equal amount of pre-trusted weight to each color (i.e. $sum_{j \in c} p_j = \frac{1}{m}$)

Run Transactions We restrict each peer i in every color c to query and download only from the peers in $succ(c)$. We thus note that for every query q, $server_i(q)$ contains only peers in $succ(c)$.

Compute Trust Values In order to compute the trust score for nodes of a given color c, we compute the stationary distribution of a modified Markov chain. We set the outgoing links from color c to be uniform over $succ(c)$, and then calculate the trust values of the nodes in c in this modified Markov chain as shown in Figure 3.

Formally, we compute the principal right eigenvector for m different matrices $\widetilde{M_c}$, one for each color in the partition. For a particular color c, let $\widetilde{d_{ij}} = d_{ij}$ if $i \notin c$ and $\widetilde{d_{ij}} = \frac{m}{n}$ if $i \in c$. Let $\widetilde{D_c}$ be the matrix $[\widetilde{d_{ij}}]$. Instead of computing

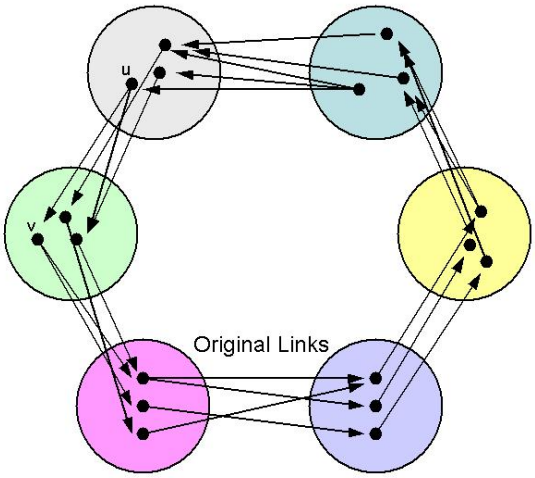

Figure 2: **Original Download Graph**
Links in the Markov chain from matrix D: a link from u to v signifies u downloaded from v successfully.

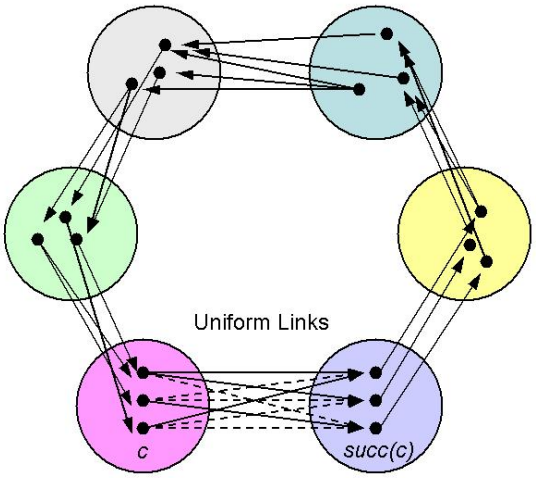

Figure 3: **Cyclic Partition Modified Download Graph for Computation of Color** c
Links in the Markov chain from matrix $\widetilde{M}_c$. Outgoing links from c are replaced with uniform links (represented by the dashed links).

the principal right eigenvector of $M = (1-\epsilon)D^T + \epsilon P$ as in EigenTrust, we compute the trust of a node i of color c as the ith component of the principal right eigenvector of $\widetilde{M}_c = (1-\epsilon)\widetilde{D}_c^T + \epsilon P$. Since the total trust in each color is $\frac{1}{m}$ in both the original and the modified Markov chain, the trust values form a probability distribution, as desired.

Distributed Implementation

We now show how to compute the m required eigenvectors in a distributed manner. While one could use standard distributed PageRank algorithms [6], we give a simple and efficient distributed algorithm which calculates all eigenvectors in comparable time to that taken to compute the EigenTrust scores. While our algorithm still computes an eigenvector, it is of a special form that can be computed quickly.

First, notice that in the Markov chain $\widetilde{M}_c$ the stationary distribution of any $i \in succ(c)$ can be calculated immediately: $t_i = (1-\epsilon)/n + \epsilon p_i$. $(1-\epsilon)/n$ comes from the uniform links of the previous color, and ϵp_i comes from the teleports into i. The stationary distribution of $succ(succ(c))$ is then just a linear combination of the stationary distributions of its parents, plus a term to account for the teleport probability.

Using this idea, the following simple and efficient algorithm can calculate the trust of every color and can be made secure using the methods outlined in [5]. For every color c, $i \in succ(c)$ initializes its distribution to $t_i = (1-\epsilon)/n + \epsilon p_i$ and sends this information to its children j, along with $\hat{d}_{ij}$. Every other node waits to be sent the distribution of its parents, then calculates $t_j = \epsilon p_i + (1-\epsilon)\sum_{k \in pa(j)} t_k \hat{d}_{kj}$. When c has calculated its trust, it stops.

This algorithm requires one linear combination per node per color, resulting in $O(m)$ linear combinations per node.

Analysis

Clearly, the trust value of peer i in round $r+1$ is independent of its report in round r, since that report is never used to calculate i's trust. This shows the following result:

Theorem 2 *The trust score defined by cyclic partitioning is myopically non-manipulable.*

We can also show that the error on the trust values is small, decreasing exponentially in m. The following corollary is proven from Theorem 6 in Appendix B, where t_i is the trust calculated according to EigenTrust and $\tilde{t}_i$ is the trust calculated by cyclic partitioning.

Corollary 1 $\sum_{i \in N} |t_i - \tilde{t}_i| \leq 2(1-\epsilon)^m$

Now, if we wish to achieve an error α given a particular teleport probability ϵ - that is, if we wish to allow the malicious nodes to gain an amount α more trust than in EigenTrust - we need only a number of colors logarithmic in $1/\alpha$.

Cut Partitioning

Having shown in the previous section an algorithm which achieves myopic non-manipulability, we now show an algorithm which achieves strong non-manipulability. We will achieve this by "cutting the loop" so that no peer can be affected by any peer whose trust it can affect. Unfortunately, our *cut partitioning* algorithm will be less effective at approximating EigenTrust and thus possibly less effective at alienating malicious peers. We make further changes only in the initialization and trust calculation steps.

Initialization As before, the center partitions the graph into m colors $C = \{1...m\}$. Again, we restrict the distribution p over pre-trusted peers to assign an equal amount of weight

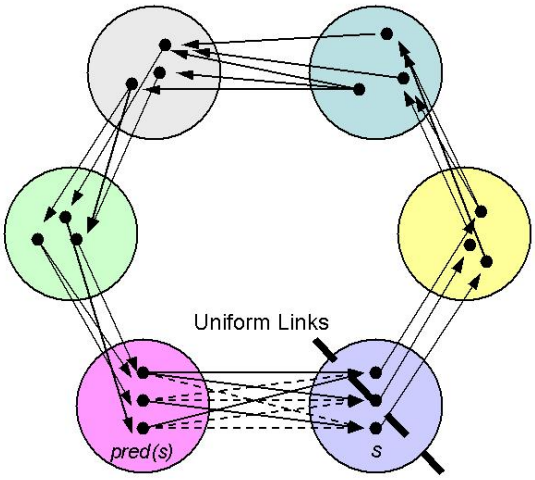

Figure 4: **Cut Partition Modified Download Graph**
Links in the Markov chain from matrix $\widetilde{M}$. Incoming links to the start color s are replaced with uniform links (represented by the dashed links). The start color s "cuts" the cycle.

to peers in each color. In the cut partitioning, however, we also choose a *starting color* $s \in C$.

Compute Trust Values In order to compute the trust score for any peer, we modify the download graph by replacing all outgoing edges of $pred(s)$ with uniform links to all peers in s as shown in Figure 4. This fixes the trust of the nodes in s, thereby making the trust of s independent of its downloads. In essence, this "cuts" the download graph at s, preventing influence from flowing across the cut.

We now need only compute the principal right eigenvector for a single matrix. Let $\widetilde{d}_{ij} = d_{ij}$ if $i \notin s$ and $\widetilde{d}_{ij} = m/n$ if $i \in s$. Let $\widetilde{D}$ be the matrix $[\widetilde{d}^r_{ij}]$. Instead of computing the principal right eigenvector of $M = (1-\epsilon)D^T + \epsilon P$ as in EigenTrust, we compute the principal right eigenvector of $\widetilde{M} = (1-\epsilon)\widetilde{D}^T + \epsilon P$.

We can compute this stationary distribution by taking the fixed trust values of the peers in s, and then propagating these forward by taking linear combinations according to the algorithm of the previous section. This is even more efficient than the previous algorithm because it only requires one linear combination per node.

Analysis

We note that the trust of peers in a color c depends on only 3 values: the trust values computed from the previous round (for colors along the directed cycle starting from s until c), the value of P, and random input (including in this the query distributions and the random selection of a single peer from among all $server_i(q)$ of a query).

Of these three values, only the trust values from the previous round are vulnerable to manipulation. Theorem 7 in Appendix C shows that the trust values from the previous round are strongly non-manipulable by proving the stronger claim that all the trust values for colors from s up to and including c are independent of the reports of peers in colors from c to s.

Furthermore, we can still achieve small error in trust values with the strong non-manipulability condition, although not as small as with the myopic condition:

Theorem 3 $\sum_{i \in N} |t_i - \tilde{t}_i| \leq \frac{2}{\epsilon m}$.

Proof The total error is $\sum_{c \in C} \sum_{v \in c} |t_v(X) - t_v(Y)| \leq \sum_{c \in C} \frac{2(1-\epsilon)^{h_{\bar{c}c}}}{m}$, using Theorem 6 from Appendix B with $\bar{c} = pred(s)$. This is a geometric series since $h_{\bar{c}c} = \{1, 2, ..m\}$ for all colors $c \in C$. Thus, $\alpha \leq \frac{1}{m} \sum_{h=1}^{m} 2(1-\epsilon)^h \leq \frac{2}{\epsilon m}$. □

We note that, although the error in trust values is small, it is concentrated in s, making the trust scores completely useless for $pred(s)$. This problem can be addressed by running several trust systems in parallel, though we reserve this discussion to the full paper.

Conclusion

There is a trade-off between approximating EigenTrust well and keeping the trust values we are approximating meaningful. A peer selecting another peer as a download source from a poorly-populated color has very little selection from among the group of peers with the requested file. Therefore the trust values will be of less use in preventing malicious peers from uploading inauthentic files. Taken to an extreme, if there is only one peer per color, trust values are useless. These considerations argue for fewer colors with more peers in each color. But the error in our trust values - that is, the maximum amount of trust malicious peers could be assigned over their trust in EigenTrust - decreases with the number of colors. Thus, increasing the number of colors increases the faithfulness of our trust values to EigenTrust but decreases the usefulness of the EigenTrust values.

Exploring ways to quantify this tradeoff is an active area of research. This includes developing an understanding of the *price of distrust*: how much worse off the overall system is due to untrustworthy users. We are currently exploring different metrics by which we can determine the usefulness of the trust values and how to apply these metrics in choosing parameters (such as the number of colors) so that the trust values are most useful.

Acknowledgements

We would like to thank our reviewers for their helpful comments and Yoav Shoham for providing valuable insights and discussion.

References

[1] H.T. Kung and C. Wu. Differentiated Admission for Peer-to-Peer Systems: Incentivizing Peers to Contribute their Resources. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[2] R. Guha, R. Kumar, P. Raghavan, and A. Tomkins. Propagation of Trust and Distrust. In *International World Wide Web Conference*, 2004.

[3] C. Ng, D. Parkes, and M. Seltzer. Strategyproof Computing: Systems Infrastructures for Self-Interested Parties. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[4] A.Y. Ng, A.X. Zheng, and M. Jordan. Link Analysis, Eigenvectors, and Stability. In *International Joint Conference on Artificial Intelligence* (IJCAI-01), 2001.

[5] S. Kamvar, M.T. Schlosser, and H. Garcia-Molina. The EigenTrust Algorithm for Reputation Management in P2P Networks. In *International World Wide Web Conference*, 2003.

[6] S. Abiteboul, M. Preda, G. Cobena. Adaptive On-Line Page Importance Computation. In *International World Wide Web Conference*, 2003.

[7] C. Buragohain, D. Agrawal, and S. Suri. A Game Theoretic Framework for Incentives in P2P Systems. In *IEEE P2P*, 2003.

[8] D. Dutta, A. Goel, R. Govindan, and H. Zhang. The Design of A Distributed Rating Scheme for Peer-to-peer Systems. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[9] Q. Sun and H. Garcia-Molina. SLIC: A Selfish Link-based Incentive Mechanism for Unstructured Peer-to-Peer Networks. Technical Report available at http://dbpubs.stanford.edu/pub/2003-46.

[10] D. Aldous. Random Walks on Finite Groups and Rapidly Mixing Markov Chains. In A. Dold and B. Eckmann, editors, Siminaire de Probabilites XVII 1981/1982. Lecture Notes in Mathematics, Vol. 986, pages 243-297. Springer-Verlag, 1983.

[11] A. Harmon. Amazon Glitch Unmasks War Of Reviewers. In *The New York Times*, Feb. 14, 2004.

[12] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. In *Stanford Digital Library Technologies Project*, 1998.

Appendix A

We wish to prove the following theorem:

Theorem 4 *If some action profile d in the single-round-game $G(\mathcal{D}, T)$ is not a 2δ-equilibrium, then there does not exist a myopically non-manipulable trust score $\tilde{T}$ s.t. $\sum_{i \in N} |T_i(d) - \tilde{T}_i(d)| \leq \delta$.*

We will proceed by proving lemma. First note that $\tilde{T}_i(\hat{d}_i, \hat{d}_{-i})$ is independent of $\hat{d}_i$ (because $\tilde{(T)}$ is non-manipulable), so we abuse notation by writing it as $\tilde{T}(\hat{d}_{-i})$.

Lemma 1 $\forall i, d_{-i}, T_i, \widetilde{T}_i, \exists d_i$ *s.t.*

$$|T_i(d_i, d_{-i}) - \widetilde{T}_i(d_{-i})| \geq \frac{1}{2}(\max_{\overline{d}} T(\overline{d}, d_{-i}) - \min_{\underline{d}} T(\underline{d}, d_{-i}))$$

Proof Given d_{-i}, let $\overline{t} = \max_{\overline{d}} T_i(\overline{d}, d_{-i})$ and $\underline{t} = \min_{\underline{d}} T_i(\underline{d}, d_{-i})$. Suppose $\tilde{T}_i(d_{-i}) \leq \frac{1}{2}(\overline{t} - \underline{t})$. Then $\max_{\overline{d}} |T_i(\overline{d}, d_{-i}) - \tilde{T}_i(d_{-i})| = \overline{t} - \tilde{T}_i(d_{-i}) \geq \overline{t} - \frac{1}{2}(\overline{t} + \underline{t}) = \frac{1}{2}\overline{t} - \frac{1}{2}\underline{t} = \frac{1}{2}|\overline{t} - \underline{t}|$. The case where $\tilde{T}_i(d_{-i}) \geq \frac{1}{2}(\overline{t} - \underline{t})$ is symmetric. □

Theorem 5 *If some action profile d in the single-round-game $G(\mathcal{D}, T)$ is not a 2δ-equilibrium, then there does not exist a myopically non-manipulable trust score $\tilde{T}$ s.t. $\sum_{i \in N} |T_i(d) - \tilde{T}_i(d)| \leq \delta$.*

Proof

We show the contrapositive. Suppose that for some d, a peer can increase its trust by at least 2δ.

$$\max_{d \in \mathcal{D}} \sum_i |T_i(d) - \widetilde{T}_i(d)|$$

$$\geq \max_{d \in \mathcal{D}} \max_i |T_i(d_i, d_{-i}) - \widetilde{T}_i(d_{-i})|$$

$$\geq \max_{\overline{d} \in \mathcal{D}} \frac{1}{2} T(\overline{d}, d_{-i}) - \min_{\underline{d} \in \mathcal{D}} T(\underline{d}, d_{-i})$$

$$\geq \max_{\overline{d} \in \mathcal{D}} \frac{1}{2} T(\overline{d}, d_{-i}) - T(d_i, d_{-i}) \geq \delta$$

Thus, the total error in approximation on d is at least δ. □

Appendix B

Consider a download graph which is partitioned into m colors, and in which a peer in any color c can only download from a peer in $succ(c)$. We wish to bound the error in trust values which arises from altering the outgoing links from some color $\bar{c}$.

Let $h_{\bar{c}c}$ indicate the number of hops along the directed cycle, starting at $\bar{c}$, and following the successor function from $\bar{c}$ until we reach c. If c is the successor of $\bar{c}$, then is $h_{\bar{c}c}$ is 1.

To prove this, we consider two coupled Markov chains: an original Markov chain X using transition probabilities from matrix M, $l_{ij} = (1 - \epsilon)d_{ij} + \epsilon p_j$ for all entries including $\bar{c}$, and a perturbed Markov chain Y that differs from X only in the variables d_{ij}, for all $i \in \bar{c}, j$, affecting outgoing links from the set $\bar{c}$. The original trust will correspond to the stationary distribution of X; the trust after altering the outgoing links will correspond to the stationary distribution of Y.

Let X_t be the location of the random walk at time t using Markov chain X and let Y_t be the symmetric notation for Y. Initially, the walk begins at two arbitrary nodes in $\bar{c}$. Both walks use the same random input, and therefore teleport at exactly the same steps to exactly the same peer. Notice, the walks are always in the same set along the cycle.

Lemma 2 *When a teleport occurs, the walks will be coupled (i.e. at the same nodes) until the walk visits $\bar{c}$ again.*

Proof The decisions whether to teleport and where to teleport are determined by the random input. Since both walks follow the same random input, when a teleport occurs they will both go to the same node. Once they are at the same place at the same time, the Markov chains are the same and they are following the same random input so it is not possible for the paths to split unless they are leaving the set $\bar{c}$. □

With slight abuse of notation, we say X_t is also the node that the random walk visits at time t, and thus $X_t, Y_t \in c$ signifies that at time t both random walks are in c.

Lemma 3 $Pr(X_t \neq Y_t | X_t, Y_t \in c) \leq (1-\epsilon)^{h_{\bar{c}c}}$.

Proof By previous lemma, no teleport occurred since the most recent time step in which both X and Y were in $\bar{c}$. Any walk from $\bar{c}$ to c that does not teleport must be at least $h_{\bar{c}c}$ hops long, and at each hop there is ϵ probability of teleporting. Therefore the probability that no teleport occurs is at most $(1-\epsilon)^{h_{\bar{c}c}}$. Since any teleport will result in $X_t = Y_t$, $Pr(X_t \neq Y_t | X_t, Y_t \in c) \leq (1-\epsilon)^{h_{\bar{c}c}}$. □

Lemma 4 $\sum_{i \in c} |t_i - \tilde{t}_i| \leq \frac{2(1-\epsilon)^{h_{\bar{c}c}}}{m}$.

Proof The distribution over events $X_\infty, Y_\infty | (X_\infty, Y_\infty \in c)$ is a coupling over the eigenvectors for Markov chains X and Y and therefore we can apply the Coupling Lemma of Aldous [10]. The probability of event $(X_\infty, Y_\infty \in c)$ is $\frac{1}{m}$, so the total error on these events is divided by m. □

Theorem 6 *For any colors c and $\bar{c}$, the error on trust in color c due to altercation of links out of $\bar{c}$ is less than $\frac{2(1-\epsilon)^{h_{\bar{c}c}}}{m}$, assuming all peers are truth-telling.*

Proof This result is immediate from the previous lemma. □

We use Theorem 6 to show that the error in the stable distribution of a particular color c is less than $\frac{2(1-\epsilon)^m}{m}$ by setting $\bar{c}$ = c, and therefore $h_{cc} = m$.

Appendix C

We use $[sc]$ to denote the links and colors along the directed cycle starting at color s and ending at color c. This includes the links out of s and the color s, and the links into c and the color c. The term $[cs]$ is similarly defined except we start at c and end at the color s. We use $[sc)$ to indicate $[sc]$ *without* the color c.

We use the subscript of a variable to denote the space of peers over which the variable is defined, and the superscript to denote the rounds over which the variable is defined. Thus, let $t^r_{[sc]}$ be the trust values of all peers in colors between s and c at time period r and $I^{<r+1}_{[cs)} = \sum_{h=0}^{r} d^h_{[cs)}$ be all the transactions reported by peers in colors between c and until s during or before round r.

Theorem 7 *$t^r_{[sc]}$ is independent of $I^{<r}_{[cs)}$.*

Proof Base Case: $t^1_{[sc]}$ is independent of $d^0_{[cs)}$ because $t^1_{[sc]}$ is based on links between s and c, which are independent of $d^0_{[cs)}$.

Inductive Step: Assume $t^{r-1}_{[sc]}$ is independent of $I^{<r-1}_{[cs)}$, then $t^r_{[sc)}$ is independent of $I^{<r}_{[cs)}$. $t^r_{[sc]}$ is only vulnerable to manipulation through $t^{r-1}_{[sc)}$. By inductive hypothesis, $t^{r-1}_{[sc)}$ is independent of $I^{<r-1}_{[cs)}$. □