A Non-Manipulable Trust System Based on EigenTrust

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

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. 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 other peer group. We show that the *total* error in the trust values decreases exponentially with the number of peer groups. In addition to theoretically guaranteeing non-manipulability, in practice our trust system performs nearly as well as EigenTrust and has better load-balancing properties.

Categories and Subject Descriptors: C.2.4 [**Computer Communication Networks**]: Distributed Systems—*Distributed applications*; J.4 [**Social and Behavioral Sciences**]: Economics

General Terms: Algorithms, Economics

Additional Key Words and Phrases: P2P, economics, mechanism design, algorithms

1. 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 [S. KAMVAR and GARCIA-MOLINA 2003; R. GUHA and TOMKINS 2004; KUNG and WU 2003] based on the recommendations of other peers. 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

Author's addresses: Abrams, Z., McGrew, R. and Plotkin, S., Computer Science Dept., Stanford University, Stanford, CA. A preliminary version of this paper [Z. ABRAMS and PLOTKIN 2004] was presented in the Workshop on Economics of P2P Systems 2004. This paper contains new results and focuses on the practical aspects of keeping peers honest in EigenTrust. Zoë Abrams is supported by NSF/CCR 0113217-001 and Robert McGrew is supported by an NSF fellowship and NSF ITR IIS-0205633.

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. The goal of these reputation systems is to isolate malicious peers from the network and encourage 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 will be unable to discriminate between selfish and malicious peers. The gaming of real-world reputation systems such as Amazon [HARMON 2004] and Kazaa show 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 score in the next round. Thus, the peer has no incentive to manipulate its recommendation. We call such a system *non-manipulable*. Our approach will be to take EigenTrust [S. KAMVAR and GARCIA-MOLINA 2003], an existing trust system, and replace it with a system that is non-manipulable but with trust scores that provably approximate EigenTrust's.

Our trust system, called *cyclic partitioning*, creates a query topology by partitioning peers into colors arranged in an ordering such that each peer only has incentives to query and download from peers in its successor color. Then, when calculating the trust score for a particular color, each peer in that color is assumed to provide neutral recommendations for peers in its successor color. We show the following results:

(1) Cyclic partitioning is a non-manipulable trust system.

(2) Cyclic partitioning approximates EigenTrust with error that decreases exponentially in the number of colors.

(3) The query routing structure of cyclic partitioning yields no additional overhead in terms of messages sent per response received, and the trust score calculation can be performed exactly and more efficiently than in EigenTrust.

(4) In simulations, cyclic partitioning performs nearly as well as EigenTrust at reducing the number of bad downloads while maintaining better load-balancing properties.

2. 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. [D. DUTTA and ZHANG] discuss issues of self-interest in the design of trust rating schemes. In citeStrat, the authors outline guiding principles for a vision of strategyproof computing in P2P systems. Buragohain et al. [C. BURAGOHAIN and SURI] 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 [SUN and GARCIA-MOLINA 2003], 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.

3. 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 the algorithm we will describe in section 6.

We assume there are two types of peers: *malicious* and *selfish*. Malicious peers desire to minimize the number of authentic downloads in the network and may collaborate to do so. Each selfish peer, on the other hand, wishes only to maximize its trust score. We model selfish peers as trust-score maximizers because, in EigenTrust, each peer is given quality of service proportional to its trust score in order to incentivize file sharing. Thus, a peer that primarily values its quality of service will seek to maximize its trust score. We assume that selfish peers do not collude to increase their trust scores.

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 peers 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.

As we are primarily interested in the problem of motivating peers to report their downloads truthfully, we assume that peers follow the protocol in their other actions. Weakening this assumption is interesting but beyond the scope of this paper. Thus, we assume that the only actions which the peers are free to choose are the reports they make to the center.

We further assume that each selfish peer i is myopic: if there is no false report of downloads which improves i's trust score in the next round, then peer i reports truthfully. Trust scores are somewhat sticky from round to round: a peer with high trust in round r is more likely to be downloaded from and thus have a higher trust in round $r + 1$. Thus, even if there is no manipulation which improves i's trust score in the next round, it is still possible for peer i to improve its trust score in future rounds, perhaps by increasing the trust of some other peer which is likely to recommend peer i in the future. We note, however, that such manipulations require a high degree of accuracy in predicting future downloads by other peers and are not likely to be a problem in real systems.

The strategic situation thus far described defines a single-round game $G_T = \langle N, \mathcal{D}, T \rangle$, the game with players N, actions of i as $\hat{d}_i \in \mathcal{D}_i$ (the set of all possible reports of downloads) 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 $T_i(\hat{d}_i^*, \hat{d}_{-i}) \geq T_i(\hat{d}_i, \hat{d}_{-i})$ 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. We will do this by making each peer indifferent between its actions $\hat{d}$. Formally, we say that a trust score T is *non-manipulable* if

$T_i(\hat{d}'_i, \hat{d}_{-i}) = T(\hat{d}'_i, \hat{d}_{-i})$ for all $\hat{d}, \hat{d}' \in \mathcal{D}$. Thus, if T is non-manipulable, honest reporting will be a DSE of the game G_T.[1]

4. THE EIGENTRUST ALGORITHM

The EigenTrust algorithm is intended to compute a trust score that 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 [A.Y. NG and JORDAN 2001] can be applied to show a bound on manipulability.

As we will present a modified version of EigenTrust in our work, we first briefly describe the EigenTrust trust score. For some query $q \in Q_i$, $j \in server_i(q)$ 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 there is a collection of *pre-trusted peers*, such as, for instance, the founders of the system, that are commonly known to be trustworthy *a priori* due to extraneous factors. The degree of pre-trust allocated to peers is captured by a commonly known distribution p over pre-trusted peers. Define $\widehat{D}$ as the matrix of reported downloads $[\widehat{d}_{ij}]$ and P as the matrix $[p_{ij} = p_j]$. We define a probability ϵ, known as the *teleport probability* for historical reasons [L. PAGE and WINOGRAD 1998], which measures how much trust the pre-trusted peers receive due to their pre-trusted status.

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 [S. KAMVAR and GARCIA-MOLINA 2003]:
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. The center sets $t_i^{r+1} = T_i(\hat{d}_i^r) = eig_i((1-\epsilon)\widehat{D^r}^T + \epsilon P)$

The EigenTrust algorithm has been shown in simulations to successfully alienate malicious peers. Also, it 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.

However, the Eigentrust algorithm also incents a selfish peer to lie about its recommendation to gain a higher trust value. To maximize its trust, a peer ought always to recommend a peer that recommended it. Consider the download graph of Figure 1 (a), and assume a uniform distribution for pre-trusted peers over all n peers: if the middle node reports a download from the right node, it will have trust $(2-\epsilon) * \epsilon/n$. If, on the other hand, it reports a download from the left node, it will have trust $1/n$. This sort of example

[1] Finally, we note that, while peers have no incentive to lie, they also have no strict incentive to tell the truth. While one could attempt to make peers strictly prefer truth-telling by a catch-and-punish scheme [D. DUTTA and ZHANG], these schemes become problematic in the presence of malicious peers, and we suggest that the criterion specified above is the best resolution of the incentive problem in the current setting.

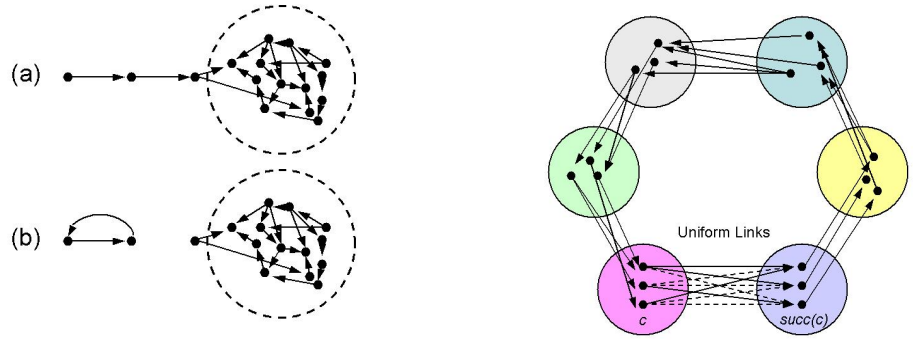

Fig. 1. **(a) Manipulation Example** Part (a) shows the actual download graph; part (b) shows a manipulation by the middle node to increase its trust. **(b)** Links in the Markov chain with uniform distribution out of color c.

is commonplace in real systems and the ratio of increase is independent of the number of peers.

5. CYCLIC PARTITIONING

We now consider modifications to the basic EigenTrust algorithm such that our new trust score is non-manipulable. We will proceed by creating a query topology and then making each peer's trust independent of his report of downloads. The query topology created 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. We change the EigenTrust algorithm as follows.

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 1 (b).

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 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.

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 1. *The trust score defined by cyclic partitioning is non-manipulable.*

We can also show that the error on the trust values is small, decreasing exponentially in m. Using the techniques of [A.Y. NG and JORDAN 2001], we can show the following theorem proved in the Appendix:

THEOREM 2. *Let t_i be the trust calculated from a particular set of downloads according to EigenTrust and $\tilde{t}_i$ be the trust calculated by cyclic partitioning. Then $\sum_{i \in N} |t_i - \tilde{t}_i| \leq 2(1-\epsilon)^m$.*

Now, in order to guarantee an error α given a particular teleport probability ϵ (in other words, 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 $1/\alpha$.

6. DISTRIBUTED IMPLEMENTATION

While cyclic partitioning may be a non-manipulable approximation of EigenTrust, it would be no use if it could not be computed effectively in a distributed fashion. In this section, we first discuss how to route queries without overhead in a color-restricted network, then show how to efficiently compute trust scores by a distributed algorithm.

Queries in the EigenTrust system are flooded throughout the network, with each peer responding to a query if it possesses the file. Naively, peers could only respond to queries from the predecessor color using the same routing architecture. In a system with c colors, however, this approach would require a factor of c more messages to reach the same number of peers who could potentially respond to the query.

However, by altering the routing topology in the graph, we can route messages without any additional overhead due to the color restriction. First, we require that peers maintain *neighbors* of only their own color. Second, we also require that peers maintain a list of *delegates* of the successor color. Now, a node which wishes to make a query first sends it to each of its delegates. The delegates forward it to their neighbors, who forward it iteratively to their neighbors. Now, no peer receives a message to which it cannot respond.

We now show how to compute the m required eigenvectors in a distributed manner. While one could use standard distributed PageRank algorithms [S. ABITEBOUL 2003], we give a simple distributed algorithm which exploits the fact that our eigenvectors are of a special form that can be computed exactly and efficiently.

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 pre-trusted weight. 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 pre-trusted weight.

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 [S. KAMVAR and GARCIA-MOLINA 2003]. 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, terminating in m rounds of communication with $O(m)$ linear combinations per node.

7. EMPIRICAL RESULTS

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.

To quantify this tradeoff, we turn to simulations. We are interested in how much worse off the overall system is due to selfish users. The metric by which we determine the success of the overall system will be the fraction of inauthentic file downloads. As we will see, even with very few colors (and therefore a large degree of error in our trust values), partitioning is still effective in limiting the number of inauthentic file downloads. Furthermore, the trust values remain useful when the number of colors is on the order of logarithm of the number of peers. We will see that malicious peers can still be alienated successfully using cyclic partitioning and that non-manipulability can be guaranteed without too much loss of effectiveness compared to EigenTrust.

In our simulations, we used the simulation environment that was used to test EigenTrust. This environment uses a power law network model, with nodes possessing files and choosing queries from a preset number of content categories. For more details, see [S. KAMVAR and GARCIA-MOLINA 2003]. All simulations are run with the same parameters as used to test EigenTrust, except that the number of peers reached by a given query was decreased (the time to live is set to 4 so as not to flood the entire network) and the number of content categories was reduced to 6 to better reflect the scale of our simulations. All data other than the trust values in the Figure 2 (b) is averaged over the last ten of 30 cycles in 4 separate runs.

7.1 Performance Against Uncoordinated Malicious Peers

First, we simulated a network with 256 selfish peers, 10 pre-trusted peers, 64 malicious peers (20%), and varied the number of colors in the partitioning from 1 (no partitioning) to 10 colors. As in threat model A used to test EigenTrust [S. KAMVAR and GARCIA-MOLINA 2003], in this simulation, malicious peers always provide inauthentic files and recommend another peer if that peer has given it an inauthentic file.

We find that, in keeping with our theoretical analysis, the total error in trust scores decreases quickly. Furthermore, the fraction of inauthentic downloads converges to that of EigenTrust with a similar speed, as shown in Figure 2 (a). Both schemes significantly reduce the fraction of inauthentic downloads compared with random node selection where no trust values are used.

Figure 2 (b) compares the global trust values of cyclic partitioning and EigenTrust. Both algorithms successfully assign a trust value of zero to the malicious peers and assign similar high levels of global trust to the high trust peers.

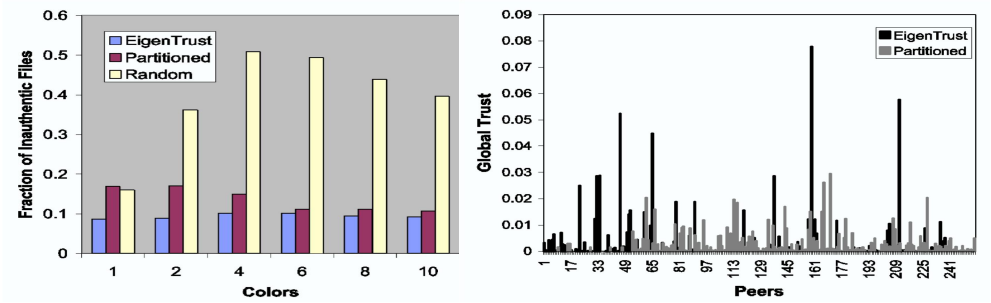

Fig. 2. **(a)** Fraction of inauthentic files versus the number of colors in the graph. **(b)** Global trust for all 256 selfish peers.

However, cyclic partitioning has an additional advantage. EigenTrust has a much higher degree of concentration in the trust values of the selfish peers than does than the cyclic partitioning trust system. This is because the division into colors disperses the trust by forcing certain peers to distribute their trust amongst a specified subset. The power law network reaches certain nodes more often than others and this property is diluted by the division into colors. Less variance in trust values among selfish peers leads to a more evenly distributed workload.

In our second simulation, we varied the number of malicious peers for several different settings of the colors and trust, as shown in Figure 3 (a). Even when half of all peers are malicious, cyclic partitioning maintains performance that is close to that of EigenTrust. The fraction of inauthentic downloads is consistently reduced by a significant portion using cyclic partitioning, as compared to when trust values are not used at all.

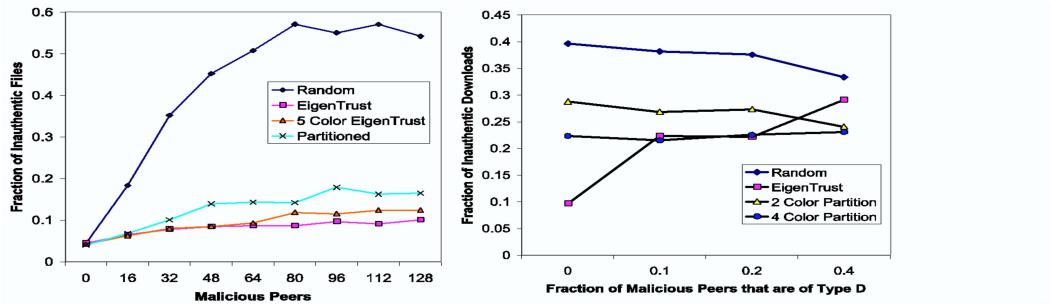

Fig. 3. Fraction of inauthentic files versus **(a)** the number of malicious peers. **(b)** the percentage of malicious peers that are of type D in the malicious collective.

7.2 Performance Against Malicious Collectives

In our third simulation, we changed the threat model to allow some malicious peers to give out authentic files to build up trust, and then recommend malicious peers, as in threat model D of EigenTrust [S. KAMVAR and GARCIA-MOLINA 2003]. This scenario is especially hard for EigenTrust - and therefore cyclic partitioning - to handle because, since these peers never give out inauthentic files, these peers are never discovered and isolated from the network.

A peer of type D responds to 5% of all queries, always answers with authentic files, and recommends every malicious peer of the successor color. Each remaining malicious peer chooses one malicious peer in its successor color at random to recommend, so that the peer's trust is not wasted (corresponding to threat model B of EigenTrust.)

The third simulation was run with 64 selfish peers, 3 pre-trusted peers, and 40 malicious peers (39%). The fraction of malicious peers of type B and D was varied as is shown in Figure 3 (b). For settings with a fraction of malicious spies up to 40%, EigenTrust and cyclic partitioning both perform better than a random strategy. Furthermore, cyclic partitioning is less dependent on the fraction of type D peers. This is because cyclic partitioning throws some information away in order to achieve non-manipulability, and therefore is less easily 'tricked' by type D peers. As can be seen, cyclic partitioning with 2 colors throws away a great deal of information and therefore exhibits a similar behavior to the random strategy as the fraction of type D peers increases. Cyclic partitioning with 5 colors, on the other hand, creates trust values much closer to those of EigenTrust. It therefore exhibits behavior similar to EigenTrust as the fraction of type D peers is increased.

Thus, against this more difficult adversary, cyclic partitioning performs comparably to EigenTrust, while maintaining non-manipulability.

8. CONCLUSION

We present a non-manipulable trust system based on EigenTrust that provably approximates the EigenTrust score and performs comparably in practice.

While EigenTrust's mathematical structure allows nice theoretical guarantees that may not carry over to other trust systems, we believe the general technique of cyclic partitioning can be fruitfully applied to guarantee non-manipulability in other trust systems as well. Evaluating the practical performance of this algorithm in other trust systems is an interesting future research direction.

APPENDIX

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 prove Theorem 2 by bounding the error in trust values which arises from altering the outgoing links from some color $\bar{c}$. The proof was first given in [Z. ABRAMS and PLOTKIN 2004] and is reproduced here for completeness.

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

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

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 [ALDOUS 1983]. 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. □

We may prove Theorem 2 using Lemma 3 and setting the color $\bar{c}$ = c and therefore $h_{\bar{c}c} = m$.

REFERENCES

ALDOUS, D. 1983. Random walks on finite groups and rapidly mixing markov chains. In *A. Dold and B. Eckmann, editors, Seminaire de Probabilites XVII 1981/1982. Lecture Notes in Mathematics, Vol. 986, pages 243-297.*

A.Y. NG, A. ZHENG. AND JORDAN, M. 2001. Link analysis, eigenvectors, and stability. In *International Joint Conference on Artificial Intelligence.*

C. BURAGOHAIN, D. AGRAWAL. AND SURI, S. A game theoretic framework for incentives in p2p systems.

D. DUTTA, A. GOEL, R. GOVINDAN. AND ZHANG, H. The design of a distributed rating scheme for peer-to-peer systems.

HARMON, A. 2004. Amazon glitch unmasks war of reviewers. In *The New York Times.*

KUNG, H. AND WU, C. 2003. Differentiated admission for peer-to-peer systems: Incentivizing peers to contribute their resources. In *Workshop on Economics of Peer-to-Peer Systems.*

L. PAGE, S. BRIN, R. MOTWANI. AND WINOGRAD, T. 1998. The pagerank citation ranking: Bringing order to the web. In *Stanford Digital Library Technologies Project.*

R. GUHA, R. KUMAR, P. RAGHAVAN. AND TOMKINS, A. 2004. Propagation of trust and distrust. In *International World Wide Web Conference.*

S. ABITEBOUL, M. PREDA, G. COBENA. 2003. Adaptive on-line page importance computation. In *International World Wide Web Conference.*

S. KAMVAR, M. SCHLOSSER. AND GARCIA-MOLINA, H. 2003. The eigentrust algorithm for reputation management in p2p networks. In *International World Wide Web Conference.*

SUN, Q. AND GARCIA-MOLINA, H. 2003. Slic: A selfish link-based incentive mechanism for unstructured peer-to-peer networks. Tech. rep., available at http://dbpubs.stanford.edu/pub/2003-46.

Z. ABRAMS, R. MCGREW. AND PLOTKIN, S. 2004. Keeping peers honest in eigentrust. In *Workshop on Economics of Peer-to-Peer Systems.*