# Resisting Sybils in Peer-to-peer Markets

Jonathan Traupman

Computer Science Division
University of California, Berkeley
jont@cs.berkeley.edu

**Summary.** We describe two techniques for reducing the effectiveness of sybil attacks, in which an attacker uses a large number of fake user accounts to increase his reputation. The first technique uses a novel transformation of the ranks returned by the PageRank system. This transformation not only reduces susceptibility to sybil attacks but also provides an intuitive and easily interpreted reputation score. The second technique, called RAW, eliminates remaining vulnerabilities and allows full personalization of reputations, a necessary condition for a sybilproof reputation system.

## 1 Introduction

Reputation systems are a key component of many large peer-to-peer and distributed applications, such as online markets, file sharing systems, and ad hoc networks. As these networks grow in size and importance, the value of a high reputation will also increase. While most users build their reputation through consistent, honest behavior, there will always be some who will attempt to manipulate the system to extract maximum benefit with minimum effort and expense. One common technique for gaming reputation systems is the sybil attack, which exploits the fact that most online applications allow the inexpensive creation of new identities. A nefarious user can easily manufacture an army of fake user accounts, the sybils, and exploit them to increase his reputation by engaging in bogus transactions and leaving undeserved positive feedback.

One proposed solution is to enforce a one-to-one correspondence between online pseudonyms and real people using a third party service created to guarantee the authenticity of pseudonyms. [8] To date, no such services have been created, and few sites implement any sort of rigorous identity screening when creating an account.

An alternative solution is to use economic effects to control the creation of sybils. If we attach a cost to creating user accounts and conducting transactions, it may be possible to render both sybil attacks and fake transactions between real users uneconomical. Bhattacharjee and Goel [2] derive the conditions necessary for a transaction fee to prevent fake feedbacks. It remains unclear, though, whether the fees needed to

*Please use the following format when citing this chapter:*

Traupman, J., 2007, in IFIP International Federation for Information Processing, Volume 238, Trust Management, eds. Etalle, S., Marsh, S., (Boston: Springer), pp. 269–284.

prevent bad behavior will be low enough so as not to discourage legitimate participation in the system. A related approach [14] makes users pay a computational cost or pass a CAPTCHA when creating an account in order to foil automated attempts to register hundreds of accounts.

If we cannot stop people from creating sybil users, then the best defense is to detect them, so that we can discount reputation information coming from sybil sources. A recent result [4] proved that any system where reputation is symmetric (i.e. where reputations are invariant under relabeling of nodes) is theoretically vulnerable to sybil attacks. Feldman et al. [6] demonstrate a scheme that uses maximum flow to form reputations in a simulated file sharing network, which is non-symmetric and effectively resists sybil attacks. Unfortunately, computing maximum flow is expensive: the fastest general algorithm requires $O(nm \log(n^2/m))$ time for a $n$-vertex, $m$-edge graph. [10] The amortized constant time approximate algorithm of [6] limits the total number of iterations of the $O(n^3)$ preflow-push algorithm [9], but they present no evidence that this approach will scale effectively to web scale networks.

The EigenTrust system [11] applies the well-known PageRank [12] algorithm to the problem of trust and reputation in peer-to-peer systems. EigenTrust's authors claim it to be resistant to not just sybils but also to collusion by otherwise legitimate users. We show in Section 2 that these claims are false and show several mechanisms for using sybils to attack EigenTrust.

We then describe a novel transformation of EigenTrust, Relative Rank, that realizes two important goals. First, it returns reputation metrics suitable for peer-to-peer markets, where both parties need to simultaneously make a decision to interact or not based on the other's reputation. Second, the reputations returned by Relative Rank resist sybil attacks.

Finally, we propose a new algorithm, RAW, that replaces PageRank within the Relative Rank framework. We prove that RAW combined with Relative Rank is secure against one main class of sybil attack and also provide a strong bound the effectiveness of the other type. Furthermore, RAW is fully personalizable: it can easily return reputations that are specific to the querying user. RAW is thus able to meet the conditions set forward by [4] as a necessary condition for a sybilproof reputation algorithm.

## 2 PageRank as a Reputation System

In order to understand the extensions to PageRank that confer sybil resistance, we must first look at the PageRank algorithm itself. This section serves as a brief summary of PageRank and of EigenTrust, an application of PageRank as a reputation system. For more details on these algorithms, we refer the interested reader to the original PageRank [12] and EigenTrust [11] papers.

### 2.1 The PageRank Algorithm

Let $G = (E, V)$ be a directed graph where every vertex has at least one outgoing edge[1]. Let $S$, the *start set*, be a vector of length $|V|$ with $||S||_1 = 1$, which defines a distribution across $V$. Let $A$ be a $|V| \times |V|$ matrix with each element $a_{ij} = 1/|\text{succ}(j)|$ if there is a link from $j$ to $i$ and 0 otherwise, where $\text{succ}(i) = \{j|(i,j) \in E\}$. The matrix $A$ is thus a stochastic matrix that represents the link structure of $G$.

Define the random walk process $\{X_t\}_{t=1...\infty}$ on $G$ with constant *damping factor* $c \in (0,1)$:

1. $\Pr\{X_0 = i\} = S_i$
2. With probability $c$, take a step such that $\Pr\{X_{t+1} = i|X_t = j\} = a_{ij}$.
3. Otherwise, restart at a random node: $\Pr\{X_{t+1} = i\} = S_i$.

The process $\{X_t\}_{t=1...\infty}$ is an irreducible, aperiodic, persistent Markov process with a finite state. By the Perron-Frobenius theorem, the process's stationary distribution, $R$, is the first eigenvector of the matrix $(1-c)S \times 1 + cA$, and can be computed with a simple iterative algorithm.

**Definition 1.** *$R_i$ is the* rank *or* PageRank score *of node $i$.*

Details of the PageRank algorithm and its applications to web search can be found in [12].

EigenTrust [11] uses PageRank as a reputation system for peer-to-peer file sharing networks. While web links are binary (either a link is present or it is not), trust relationships are described using a range of values, both positive and negative. When constructing the $A$ matrix, EigenTrust therefore uses a more complex normalization procedure. A user $i$ defines his satisfaction with user $j$, $s_{ij}$ as:

$$s_{ij} = \text{sat}(i,j) - \text{unsat}(i,j)$$

where $\text{sat}(i,j)$ and $\text{unsat}(i,j)$ represent respectively the number of satisfactory and unsatisfactory interactions that user $i$ has had with user $j$. The elements of the $A$ matrix are defined by:

$$a_{ij} = \frac{\max(s_{ij}, 0)}{\sum_k \max(s_{ik}, 0)}$$

Two important consequences of this normalization process are (1) that the random walk now chooses an outgoing link with probability proportional to the user's satisfaction instead of uniformly and (2) that negative satisfaction ratings are essentially discarded: negative trust is treated the same as no trust.

The creators of EigenTrust propose two decision procedures to use when applying this reputation information. In the first procedure, the user always picks the partner who has the highest EigenTrust score. In the second, the user chooses randomly with probability proportional to the potential partners' scores.

[1] In real networks, some nodes may not have outgoing links. There are several possible solutions to this problem: we could trim out nodes that link to no one, or we could add a link from a node to all the start set nodes. In our implementation, we do the latter.

### 2.2 Problems with EigenTrust

Despite the optimistic claims in [11], EigenTrust has a number of problems as a reputation algorithm for peer-to-peer markets:

**EigenTrust is vulnerable to collusion and sybils.** While [11] claim to demonstrate that EigenTrust is robust to collusion, their evaluation is flawed. Consider the simple collusion scenario where a set of users all agree to form a "feedback clique:" they each leave a maximally positive rating for all other members of the clique. Under such an attack, our tests have shown that each member's rank increases. Furthermore, even a single user can construct a network of sybils that will increase his rank as shown in the next section.

**EigenTrust does not have a clear decision procedure.** In peer-to-peer markets, users need to be able to look at a potential partner's reputation and decide whether to interact or not. EigenTrust scores are more or less a measure of the degree to which a node is "linked in" to the rest of the graph, and this score grows roughly linearly with the number of transactions. Consequently, the decision procedures proposed by [11] are flawed: they tend to select more experienced, but not necessary more trustworthy, partners.

**EigenTrust does not use negative feedback.** Most online markets allow both positive and negative feedback. EigenTrust's strategy of discarding this negative information is sub-optimal. Because EigenTrust scores grow linearly with the number of positive links and ignore the negative ones, a user with a fairly high rate of negative feedback can still see unbounded grown in his EigenTrust score.

**EigenTrust is vulnerable to attacks by users in the start set.** The vertices with positive probability in the start set distribution fill a special role in PageRank-like algorithms. As the starting point for the random walk, these nodes are the source of all authority in the graph. In classical implementations of PageRank, this start set contains all top level domains, weighted uniformly. In EigenTrust, the start set is a set of "trustworthy" nodes established by the management of the reputation system. In both cases, this start set remains the same for all queries, resulting in a symmetric reputation function, which is provably not sybilproof [4]. While the cost of top-level domains [5] and careful selection of trustworthy nodes in EigenTrust can raise the cost and reduce the effectiveness of sybil attacks, they cannot be eliminated. Furthermore, the power wielded by start set members is an invitation for corruption.

Fortunately, none of these pitfalls is insurmountable. We spend the remainder of this report examining these weaknesses and their solutions in detail.

## 3 Sybil Attacks

Broadly speaking, there are two ways in which sybils can be helpful: the attacker can use use them to increase his own reputation or he can use a sybil, rather than his main identity, to conduct transactions with other users. We concentrate first on attacks designed to increase the attacker's reputation. With PageRank or EigenTrust, if an attacker can alter the random walk process to increase the amount of time it

spends at his node, then he can increase his rank. We assume that the only way an attacker can affect the random walk is by engaging in fake transactions with sybils, thus adding links among his main node and the sybils. It is also possible to use the sybils to engage in transactions with other users, but this tactic is counter-productive if the attacker's goal is to increase his main node's reputation:

**Proposition 1.** *Let $G = (E, V)$ be the trust graph excluding the attacker node and all its sybils. Let $G_a = (E_a, V_a)$ be the graph of the attacker node $v_a \in V_a$ and its sybils $\{s_0, \ldots, s_n\} \subset V_a$. Let $G_C = (E_C, V_C)$ be the complete graph with $V_C = V \cup V_a$ and $E_C = E \cup E_a \cup \{(i, j) : i \in V, j \in V_a\}$.*

*The rank of the attacker $v_a$ is maximized when all edges $(i, j)$ between nodes in $G$ and nodes in $G_a$ are connected to $v_a$.*

*Proof (informal).* Consider incoming edges $(i, j)$ where $i \in V$ and $j \in V_a$. If $j = v_a$, then on each transit of $(i, j)$, the random walk will visit $v_a$, increasing its rank. However, if $j \neq v_a$, then the probability that the random walk visits $v_a$ after transiting $(i, j)$ is strictly less than one. So, to maximize its rank, an attacker would want to have edges incoming from $G$ to $G_a$ to go to his main node, not one of the sybils.

Outgoing edges $(i, j)$, where $i \in V_a$ and $j \in V$, fall under a similar argument. If $i = v_a$, then all random walks exiting $G_a$ must first visit $v_a$ increasing its rank. If $i \neq v_a$, then it is possible for a random walk to exit $G_a$ without visiting $v_a$. So to maximize its rank, the attacker should have all outgoing edges connected to $v_a$.

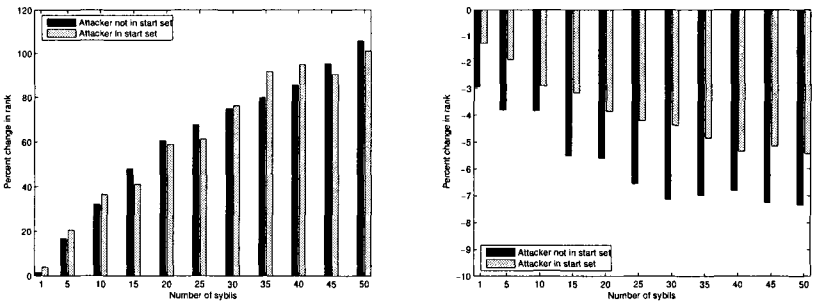
A more formal proof of this result can be found in [3].

### 3.1 Attack Types

While Proposition 1 shows that an attacker cannot increase his reputation through cleverly choosing sybils to engage in transactions, it is nevertheless possible to engineer a network of sybils that increases the attacker's score. Informally, a node's EigenTrust score is the ratio of visits to the node to the total number of steps in the process, so there are two strategies for increasing it: increase the number of visits to the node or make fewer visits to other nodes.

A *Type I* attack uses sybils to redirect the random walk back at the attacker's node, increasing the number of visits to it. A simple configuration that implements this attack creates $N$ sybils and adds both in- and outgoing links between each sybil and the attacker. Provided the attacker has no other outgoing links (or $N$ is much larger than the number of outgoing links), once the process enters the sybil network, it will spend approximately half its time visiting the attacker until it resets to a new start set node.

In the *Type II* attack, the attacker links to each sybil but does not link back to his main node: each sybil is a dead end. This attack forces the process to restart at a start set node more frequently, preventing visits to nodes outside the sybil network. Sybils are not strictly necessary in this attack: an attacker with no outgoing links at all also achieves the same end. However, if the attacker has outlinks to non-sybil nodes, he




(a) Type I (b) Type II

Fig. 1: Effectiveness sybil attacks against the EigenTrust reputation system.

will need a significantly larger number of links to dead-end sybils to cause a high restart probability. While forcing a reset of the process does prevent visits to other nodes after it sees the attacker, the low probability of returning to the attacker render it unclear whether this attack is of much use. In practice, we have seen little benefit to using this attack, but we include it for completeness.

The *Type III* attack, which uses the same network topology as the Type II attack, has a different goal. Instead of increasing the attacker's reputation, the purpose of this attack is to create sybils with high reputations that can then be spent engaging in uncooperative behavior without affecting the attacker's primary reputation. Once a negative feedback diminishes a sybil's reputation, the attacker simply discards it.

### 3.2 EigenTrust is not Sybilproof

To investigate the effect of these three sybil attacks on the EigenTrust algorithm, we implemented them in our marketplace simulator (described in detail in [13]). We measure the effectiveness of the first two attack types by looking at the percentage change in reputation. For the Type III attack, we simply look at the mean reputation of the created sybils. For each test, we ran 10 independent simulations, each with 10 attackers with the final results obtained by taking the mean of all 100 attackers.

Figure 1 shows the results of this test. The Type I attack is clearly effective: even a single sybil causes a measurable increase in reputation and 50 sybils allows the attacker to more than double his reputation. The effectiveness of this attack strictly increases with the number of sybils, although the incremental benefit is less with more sybils. The attack is roughly equally effective whether the attacker belongs to the start set or not; however, the members of the start set begin with much higher reputations, so the absolute increase is greater.

The Type II attack (Figure 1b) is not effective at all, with sybils causing a decrease in reputation at all levels. It is slightly less ineffective if the attacker is a member of the start set, since the chances of returning to the attacker after restarting

a random walk is much higher. While of some theoretical interest, this attack does not appear to be of much concern for practical systems.

It is difficult to evaluate the effectiveness of the third attack (see Figure 6 below) because, as we discussed in Section 2.2, it is unclear exactly what constitutes a *good* or *bad* reputation under EigenTrust. However, sybils do receive a positive reputation, though more sybils means each sybil's reputation is slightly lower. More troubling is that the reputations of sybils created by a start set member are, on average, nine times higher than those created by a non-member. Since the configuration of sybils in the Type III attack is identical to that of the Type II attack, we note that a start set member can trade off a small (roughly 5%) decrease in his main identity's reputation in order to create an army of relatively high reputation sybils.

## 4 Relative Rank: PageRank for Markets

We now introduce our technique of *Relative Rank*, a transformation of EigenTrust scores with several desirable properties:

- Relative Rank has a clear decision procedure. Honest users, regardless of their experience, receive high Relative Rank scores, while dishonest ones receive low scores, permitting users to use a simple constant threshold.
- Relative Rank uses negative feedback. A user with a steady rate of bad behavior will have a lower Relative Rank than one whose behavior is consistently honest.
- Relative Rank resists sybil attacks. For users that are not members of the start set, Relative Rank does not increase with either Type I or Type II sybil attacks. Furthermore, the sybils created in a Type III attack have reputation too low to reliably engage in transactions on the attacker's behalf.

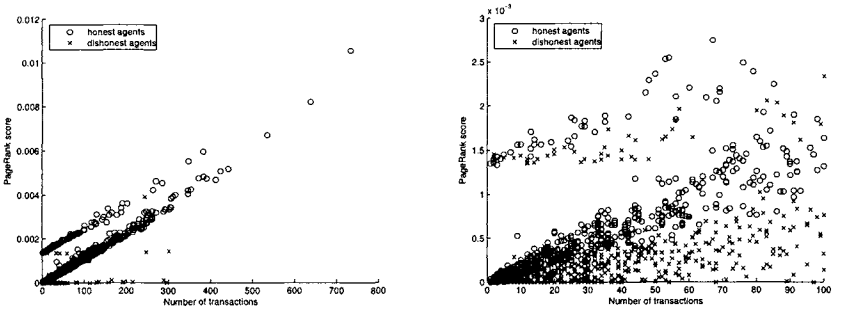
### 4.1 Relative Rank Defined

The original motivation for Relative Rank was to transform PageRank into a reputation system suitable for use in peer-to-peer markets. In typical markets, potential buyers and sellers examine each others' reputations and try to decide whether or not it is safe to interact. In systems like Percent Positive Feedback, used by eBay, a high reputation corresponds to a high estimated success rate, allowing users to apply a simple threshold when deciding whether or not to interact.

Under EigenTrust, a user's score increases with the number of positive feedbacks received, not with the success rate of the user. Additionally, users in the start set begin with much higher rank than non-members. However, enlarging the start set to include all users allows a new, trivial sybil attack. [5]

Figure 2 plots EigenTrust score against the number of transactions for all users in two simulated markets. In the first market, we use a bimodal distribution of agent honesty:[2] 95% of users are honest and behave honestly an average of 98% of the

[2] We use the term "honesty" as a shorthand for "probability of acceptable performance." As suggested by [1], we do not try to assess user motivation or make a distinction between incompetence and malice.




(a) Non-uniform (b) Uniform

Fig. 2: EigenTrust score vs. number of transactions for all users in two simulated markets.

time. The remainder average honest behavior only 2% of the time. We believe that this distribution captures the essence of real networks where users tend to either play by rules or cheat all the time, and not use some mixed strategy. The overall mean honesty in this market is 93.2%. In the second market, user honesties are distributed uniformly. Honest users are those that behave honestly at least as often as the mean.

Examining Figure 2a, we see four major regimes:

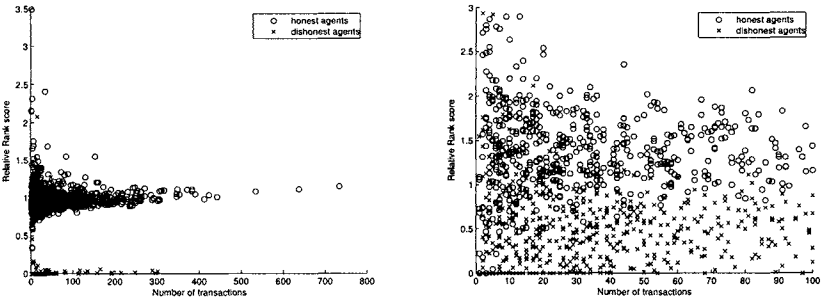
1. Honest agents whose rank follow a line with positive slope and intercept 0.0015
2. Honest agents whose rank follow a line with positive slope and intercept 0
3. Dishonest agents whose rank lies around 0.0015, regardless of experience
4. Dishonest agents whose rank lies around 0, regardless of experience

Similar patterns exist in the uniformly honest market (Figure 2b) as well.

Encouragingly, the rank of dishonest agents behaves differently than that of honest ones. However, it is clear that a simple threshold won't work very well: a threshold less than 0.0015 will miss many dishonest users, while one much greater than 0 will classify a large number of honest agents incorrectly. Groups 1 and 3 represent users that belong to the start set and the other groups consist of non-members. However, even if we divide the users based on start set membership, any threshold we set will likely exclude a large portion of users with low experience.

If we plot only users of a fixed level of honesty, we observe that the plotted points roughly follow a ray beginning at the origin (or at $(0, 0.0015)$ for start set members) and extending into the first quadrant. The angle this ray forms with the x axis is proportional the user's honesty. This observation forms intuition behind the Relative Rank algorithm:

1. Run EigenTrust.
2. Separate start set members from other users.
3. For each feedback count $k$, including *both* positive and negative feedback, find the non-start-set user $i_k$ that has the highest observed rank, $r_{i_k}$ among users who have received $k$ feedbacks.




(a) Non-uniform (b) Uniform

Fig. 3: Relative Rank versus number of transactions in the two example markets.

4. Fit a line to the pairs $(k, r_{i_k})$ and obtain a slope, $\beta_{\bar{S}}$, and intercept, $\alpha_{\bar{S}}$.
5. Repeat steps 3 and 4 for start set members to obtain a separate intercept and slope, $\alpha_S$ and $\beta_S$.

For a non-start-set user $i$ with $k$ feedbacks, define the *Relative Rank score* as:

$$s_i = \frac{r_i - \alpha_{\bar{S}}}{\beta_{\bar{S}} k}$$

The same definition holds for start set members, except that $\alpha_S$ and $\beta_S$ are used.

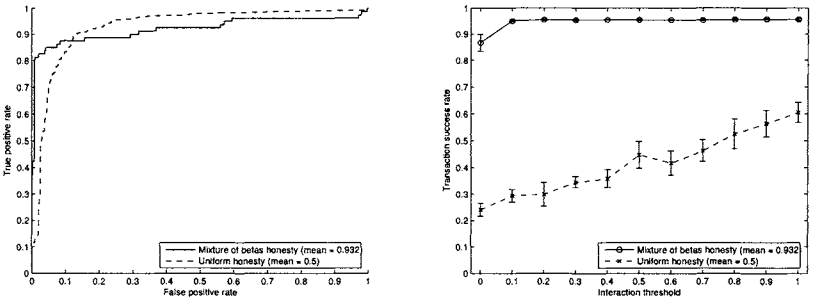
Similar plots of Relative Rank versus number of transactions for the two example markets can be found in Figure 3. Clearly, a simple linear separation between honest and dishonest users appears to be a good approach in both of these markets.

### 4.2 Reputation System Performance

Before we look at its performance with sybils, we examine how well Relative Rank serves as a reputation metric. Certainly, the ability to resist sybils is moot if the system cannot sort out good users from bad.

Figure 4a presents a ROC curve that illustrates the trade-off between detecting dishonest users and incorrectly labeling honest users as dishonest when using Relative Rank with a simple fixed threshold in our two example markets. The area under this curve is considered a good non-parametric estimation of a classification algorithm's performance, with an ideal system having area 1. For Relative Rank, the area under the curve is .9306 for the market with uniform honesty and .9212 for the market with a bimodal honesty distribution. In both cases, we define an honest user as one whose rate of successful transactions is equal or greater to the mean. If we relax this definition somewhat so that an honest user is one that behaves correctly 90% of the time, the area under the curve for the bimodal market increases to 0.996.

In Figure 4b, we measure the transaction success rate (the percentage of transactions where both parties behave honestly) in the example markets. We compared




(a) ROC curve

(b) Transaction Success Rate

Fig. 4: Performance of the Relative Rank algorithm in our example markets. Error bars in (b) indicate standard deviation across ten trials.

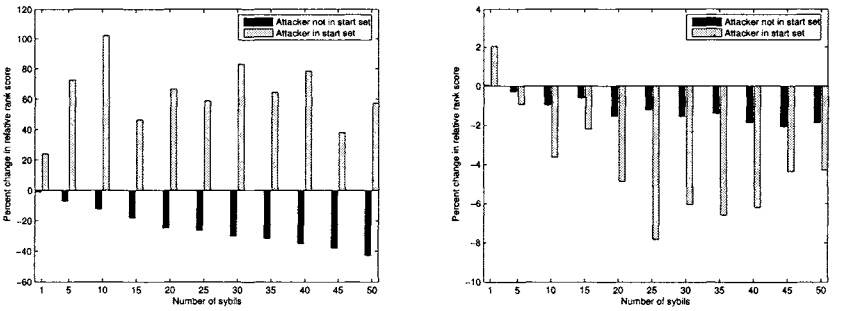
the market's performance with several different interaction thresholds (the minimum reputation an agent must have before being allowed to interact). Even with a relatively low interaction threshold, Relative Rank was able to roughly halve the number of failed transactions in both markets.

Relative Rank nearly perfectly separates the two modes in the bimodal market: with a threshold of 0 (all users always interact) the observed transaction success rate was .866, very close to the expected rate of .869. However, with Relative Rank and a moderate positive threshold (0.4–0.6), the success rate increased to .956, just slightly less than the .960 rate expected if only the honest users were permitted to operate. However, Relative Rank seems less capable of making fine discrimination between agent honesties: increasing the threshold further does not provide a significant benefit. This is not unexpected: with roughly equal honesty and experience, there will be some variation in users' Relative Rank scores depending on the local topology of the graph in which they operate. We do not view this as a problem — there is ample evidence that suggests that a bimodal distribution of users with a mostly honest majority and a dishonest minority is a reasonable model of real user behavior. Furthermore, it is exactly this sensitivity to graph structure that gives Relative Rank its resistance to sybil attacks.

### 4.3 Relative Rank and Sybils

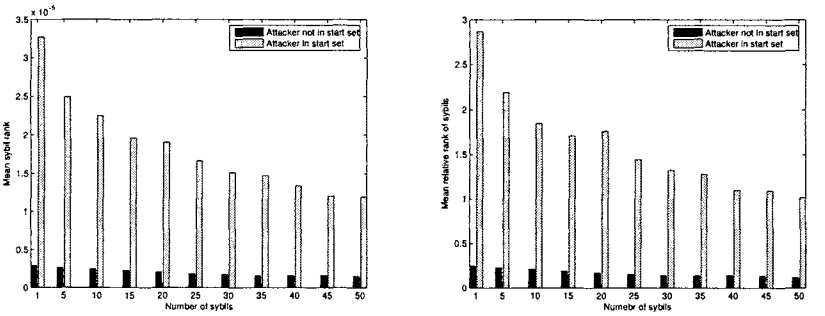
Now that we have established that Relative Rank is a useful reputation algorithm for peer-to-peer markets, we examine its behavior under the three sybil attack scenarios described in Section 3.1. The results of this experiment are shown in Figure 5. Comparing these graphs with the results for EigenTrust (Figure 1), we see that Relative Rank is significantly more resistant to sybil attacks.

The Type I attack (Figure 5a) is completely ineffective for users that do not belong to that start set but remains a viable means for start set members to increase their reputations. The Type II attack (Figure 5b) is, once again, more or less useless:




(a) Type I (b) Type II

Fig. 5: Performance of Relative Rank under the sybil attack scenarios described in Section 3.1 in the bimodal example market.




(a) EigenTrust (b) Relative Rank

Fig. 6: Performance of (a) EigenTrust and (b) Relative Rank under the Type III attack.

nearly all attackers see their Relative Rank fall with sybils. One exception is for start set nodes with only one sybil, which gives a very small reputation increase, but this small increase is of little practical benefit to the attacker.

Since, unlike EigenTrust, we have an interaction decision procedure for Relative Rank, we can analyze the impact of the Type III attack (Figure 6b) more thoroughly. The results of the previous section suggest that a good interaction threshold for this example market is around 0.5. All of the sybils created by non-start set users are thus useless: their reputation is below the interaction threshold, so it is unlikely that the attacker can use them to engage in any transactions.

However, sybils created by start set members have very high reputations. If used to commit fraudulent transactions, $f$ negative feedbacks will reduce a sybil's Relative Rank by a factor of $1/f$. An attacker can thus create a large number of sybils with only minimal effect on his main identity's reputation and conduct a large number of

fraudulent transactions (e.g. approximately 3 transactions per sybil with 25 sybils) before the sybils' reputations are expended.

While initially envisioned as merely a way of adapting EigenTrust to peer-to-peer markets, Relative Rank had the unexpected benefit of increased resistance to sybil attacks, at least by attackers that do not belong to the start set. However, it is still vulnerable to abuse by start set members. We also cannot prove this sybil resistance: it appears to be generally true, but may simply be an artifact of our choice of simulation parameters.

# 5 The RAW Algorithm

To address the few remaining concerns with Relative Rank, we introduce RAW, a PageRank-like algorithm with two important properties:

1. Provable immunity to Type I attacks and a provable bound on the effectiveness of Type II sybil attacks.
2. Asymmetric, personalized reputations, which render attacks that rely on start set membership ineffective.

RAW does not replace Relative Rank; rather, it replaces the PageRank implementation within the core of the Relative Rank framework. The combination of RAW with Relative Rank achieves our goal of a highly sybil resistant reputation system for peer-to-peer markets.

## 5.1 Definition of the RAW Algorithm

The setup for RAW is the same as for PageRank: we have a directed graph, $G = (E, V)$, representing the users and their trust relations as well as a start set, $S$ and constant damping factor, $c \in (0, 1)$. The RAW process, $\{(X_t, H_t)\}_{t=1\ldots\infty}$ is a random walk on the graph that proceeds according to the following rules:

1. $H_0 = \emptyset$, $\Pr\{X_0 = i\} = S_i$.
2. With probability $c$, set $H_{t+1} = H_t \cup \{X_t\}$ and take a step such that $\Pr\{X_{t+1} = i | i \in H_t\} = 0$ and $\Pr\{X_{t+1} = i | X_t = j, i \notin H_t\} = a_{ij} / \sum_{k \in \text{succ}(j) \setminus H_{t+1}} a_{kj}$.
3. Otherwise, $H_{t+1} = \emptyset$ and $\Pr\{X_{t+1} = i\} = S_i$.

**Definition 2.** *If $R$ is the length $|V|$ vector describing the stationary distribution of $X_t$ in the process $\{(X_t, H_t)\}_{t=1\ldots\infty}$ defined above, then $R_i$ is the* RAW score *of node $i$.*

This process is very similar to the one used to define PageRank with one important difference: the process cannot visit the same node more than once between resets to a random start set node. This property is the key to RAW's sybil resistance. No configuration of edges can cause the process to revisit a node, so the Type I attack is impossible by definition.

RAW behaves very similarly to PageRank in the absence of Sybils and can be used as a "drop-in" replacement in EigenTrust, Relative Rank, or any other system that uses PageRank.

### 5.2 Implementation and Personalization

The addition of history obviously renders the RAW process non-Markov, so simple close-form or iterative formulations of its stationary distribution are not readily apparent. For the experiments in this paper, we use a Monte Carlo implementation that directly simulates the random walk process.

For deployment in a web-scale marketplace, it will be necessary to efficiently scale up this implementation from thousands to millions of nodes. Similar techniques have been proposed for Personalized PageRank web search systems [7], and these systems can be readily adapted to computing RAW scores instead.

A key benefit of this implementation of RAW is that it can be fully personalized. To accomplish this, we create a collection of start sets, each with only a single member. We then run the Monte Carlo simulation of the RAW random walk to build a "fingerprint" of ranks for that user — in essence the RAW scores using just that single node as the start set. These fingerprints are stored in a database for easy access.

At query time, the user chooses which nodes to include in the start set and looks up the RAW scores of the target in the fingerprint database. The user then constructs a personalized RAW score by taking the (optionally weighted) average of the queried fingerprint values. In this way, the user creates the start set dynamically for each query. A proposition in [7] proves that a start set built up in this fashion is equivalent to a start set chosen in the standard way.

In a practical system, the market administration will want to build a fingerprint database large enough to offer a user a wide choice of start set nodes, yet small enough to make the Monte Carlo RAW calculation tractable. Users then choose unique subsets of this "meta-start set" for individual queries. Provided the meta-start set is large enough, a user will be able to find a sufficiently large start set that does not include either the node whose reputation is being queried or any of its immediate neighbors, drastically reducing the effectiveness of sybil attacks that rely on start set membership or proximity.

### 5.3 RAW and Sybils

The proof of RAW's immunity to Type I attacks is by definition: RAW prohibits multiple visits to the same vertex between resets to a start set node, so any configuration of sybils that attempts such a redirection will fail. Obviously, this immunity to Type I attacks also carries over to RAW Relative Rank: feedback from sybils cannot increase the RAW score, but it does increase the feedback count, thus decreasing Relative Rank score.

Type II attacks are theoretically possible against RAW; however, we can prove a tight bound on their effectiveness.

**Proposition 2.** *Let $r_i$ be the RAW rank of a user, $i$, without any sybils and let $r_i'$ be the RAW rank of the same user after creating sybils in a Type II configuration. If $c$ is the chosen damping factor, then the effectiveness of the attack is bounded by* $\mathbb{E}[r'/r] < (1 - c^3)^{-1}$.

*Proof.* We consider the worst case: there is a single start set node, $s$, that is the source of all random walks. It is connected directly to $i$ and to no other nodes. This configuration maximizes the number of visits to $i$, because $i$ lies along the path of all walks of length 2 or more. The attacker has connections to $n$ non-sybil nodes.

The expected number of visits to $i$ on each walk is simply the damping factor $c$. The expected walk length given a visit to $i$ is $1+c+c^2(1+l)$, where $l$ is the expected length of a random walk in the non-sybil portion of the graph. So, the expected rank of $i$ without sybils is:

$$\mathbb{E}[r] = \frac{c}{1+c+c^2(1+l)}$$

When $i$ creates $m$ sybils in a type II configuration, the walk transitions from $i$ to a sybil with probability $m/(m+n)$, so the expected rank with sybils is:

$$\mathbb{E}[r'] = \frac{c}{1+c+c^2(1+\frac{m}{m+n}l)}$$

If we take the limit as $m \to \infty$, we get that:

$$\mathbb{E}[\frac{r'}{r}] = \frac{1+c+c^2(1+l)}{1+c+c^2}$$

If the random walk never hits a dead end, then $\mathbb{E}[l] = c/(1-c)$. Because dead ends are possible, $\mathbb{E}[l]$ is strictly less than this value. Making this substitution for $l$ gives us our bound.

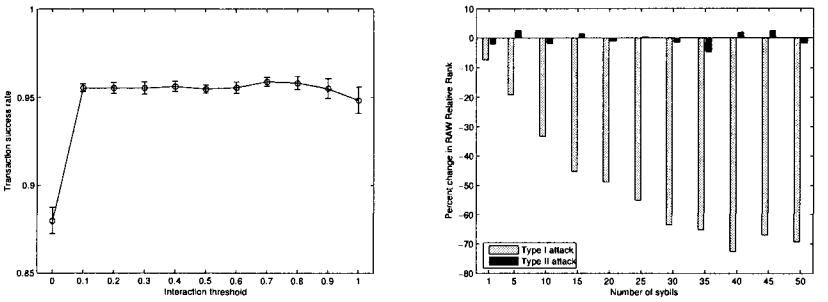
For the choice of $c = 0.85$ used in our experiments, the maximum increase in reputation with an attack of this type is approximately 2.6. We can also solve the above equation for $c$ given a desired bound on $r'/r$.

In practice, attacks of this form are even less effective because there are many start set nodes, making the probability of returning to the attacker extremely low. Furthermore, with personalization, the membership of the start set can change arbitrarily often, making it essentially impossible to consistently gain a practical increase in reputation.

### 5.4 Results

Figure 7a plots the transaction success rate against the interaction threshold for RAW Relative Rank in our simulated market. Compared to standard Relative Rank (Figure 1), there are few differences. Both systems are about equally effective at preventing failing transactions. However, the RAW version experiences a slight reduction in transaction success with high ($> 0.8$) interaction thresholds, due to higher score variances introduced by the Monte Carlo implementation. Once again, a moderate interaction threshold of around 0.5–0.7 makes the best trade-off between preventing failed transactions and not deactivating too many honest agents.

Performance with sybils (Figure 7b) is as predicted by theory. Neither Type I nor Type II sybil attacks achieve any practical measure of success in increasing the




(a) Transaction Success (b) Type I & II Attacks

Fig. 7: Evaluation of RAW Relative Ranks used as a reputation system.

attacker's RAW Relative Rank. Sybils created in a Type III attack have RAW relative ranks in the 0.25–0.35 range, similar to what we saw with standard Relative Rank for non-start set members. However, with RAW Relative Rank, the "start set" disappears as a concept, so it is not possible for an attacker to exploit his start set membership to launch a successful Type III attack.

## 6 Conclusion

In this report, we presented two techniques that make considerable progress towards the goal of a fully robust reputation system for peer-to-peer markets. The Relative Rank algorithm takes the widely studied PageRank algorithm and adapts it for use as a marketplace reputation system. It transforms users' EigenTrust scores, which are dependent on their experience level, into a reputation metric that can be easily thresholded against for making trust decisions. Furthermore, it incorporates negative feedback so that users must maintain a high degree of honesty in order to be judged worthy of interacting. Finally, Relative Rank is more resistant to sybil attacks than PageRank: for non-start set users, all three of the sybil attacks we identified fail.

The RAW algorithm replaces PageRank within the Relative Rank framework resulting in several key benefits. Unlike PageRank, RAW is, by definition, invulnerable to Type I sybil attacks. Type II attack success can be bounded, and in practice is far lower than even the bound suggests. Finally, RAW is completely personalized: the querier can choose the start set, so reputations are asymmetric. Combined with Relative Rank, RAW becomes a reputation algorithm with a simple decision procedure for peer-to-peer markets, resistance to all three classes of sybil attacks, and no opportunity for corruption by start set members.

## Acknowledgments

The research contained in this report was performed in collaboration with Prof. Robert Wilensky of U.C. Berkeley. While he was unable to participate in writing this report, we would like to acknowledge his many contributions to it.

This work was supported in part by TRUST (The Team for Research in Ubiquitous Secure Technology), which receives support from the National Science Foundation (NSF award number CCF-0424422) and the following organizations: AFOSR (#FA9550-06-1-0244) Cisco, British Telecom, ESCHER, HP, IBM, iCAST, Intel, Microsoft, ORNL, Pirelli, Qualcomm, Sun, Symantec, Telecom Italia and United Technologies.

## References

1. K. S. Barber, K. Fullam, and J. Kim. Challenges for trust, fraud and deception research in multi-agent systems. *Trust, Reputation, and Security: Theories and Practice*, pages 8–14, 2003.
2. R. Bhattacharjee and A. Goel. Avoiding ballot stuffing in ebay-like reputation systems. In *Proc. SIGCOMM '05 P2P-ECON Workshop*, 2005.
3. M. Bianchini, M. Gori, and F. Scarselli. Inside pagerank. *ACM Transactions on Internet Technology*, 5(1), February 2005.
4. A. Cheng and E. Friedman. Sybilproof reputation mechanisms. In *Proc. SIGCOMM '05 P2P-ECON Workshop*, August 2005.
5. A. Clausen. The cost of attack of pagerank. In *Proc. International Conference on Agents, Web Technologies and Internet Commerce (IAWTIC)*, 2004.
6. M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust incentives for peer-to-peer networks. In *Proc. ACM E-Commerce Conference (EC'04)*, May 2004.
7. D. Fogaras, B. Rácz, K. Csalogány, and T. Sarlós. Toward scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. *Internet Mathematics*, 2(3):333–358, 2005.
8. E. Friedman and P. Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10:173–199, 2001.
9. A. V. Goldberg. *Efficient Graph Algorithms for Sequential and Parallel Computers*. PhD thesis, MIT, 1987.
10. A. V. Goldberg and R. E. Tarjan. A new approach to the maximum-flow problem. *J. ACM*, 35(4):921–940, 1988.
11. S. D. Kamvar, M. T. Scholosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proc. WWW 2003*, May 2003.
12. L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. Technical Report 1999-66, Stanford University, 1999.
13. J. Traupman and R. Wilensky. Robust reputations for peer-to-peer marketplaces. In *Proc. 4th International Conference on Trust Management (iTrust)*, 2006.
14. L. von Ahn, M. Blum, and J. Langford. Telling humans and computers apart (automatically). Technical Report CMU-CS-02-117, Carnegie Mellon University, School or Computer Science, 2002.