A Priori Trust Vulnerabilities in EigenTrust

Rob Jansen
jansen@cs.umn.edu

Ted Kaminski
tedinski@cs.umn.edu

Fedor Korsakov
korsa004@umn.edu

Alexander Saint Croix
saintx@umn.edu

Daniel Selifonov
selifon@cs.umn.edu

December 22, 2008

Abstract

We review the motivations underlying the design of the EigenTrust [4] algorithm for trust management in P2P file sharing networks, then illustrate through a simulated P2P network how EigenTrust's reliance on pre-trusted peers undermines its other security requirements and can lead to severely compromised networks. We then explore a potential alternative to the use of pre-trusted peers, and demonstrate that EigenTrust can work without these vulnerable authorities.

1 Introduction

As online peer-to-peer (P2P) file sharing networks continue to gain widespread acceptance, the risk of computer viruses and worms using them as a propagation medium continues to grow. Under present conditions, users of such networks face the prospect of "a market for lemons," where malicious or damaged files continue to replicate as there is no incentive to remove them[1]. The prevalence of inauthentic files discourages participation by trustworthy hosts, over time decreasing the quality of the network, and leading to cycle of deterioration. Various methods have been proposed to mitigate this risk and provide users of P2P file sharing networks with tools to accurately predict the trustworthiness of peers on the network by adding reputation management systems to contemporary P2P networks [4, 8, 9].

Reputation management systems are tools that attempt to record, communicate, and even predict the trustworthiness of peers in a network with the goal of allowing users to see past their own direct experience. Examples of common reputation management systems currently in use include the transaction feedback mechanisms present in eBay and Amazon.com, as well as the comment ranking system present in Slashdot[5].

A P2P reputation system should allow users of a P2P file sharing service to advertise the quality of their transactions with other users in order to help other users find high quality peers to engage with in file sharing. This should ideally punish and discourage untrustworthy or poorly performing peers, and reward highly performing or trustworthy peers. As a result, non-malicious users should have an incentive to prune bad files from their collection, and

malicious users should be marginalized from the network. The system must also be prepared to handle large collectives of actively malicious peers.

Kamvar, Schlosser, and Garcia-Molina describe such a reputation system, which they call EigenTrust [4]. EigenTrust was shown in P2P network simulations to significantly reduce the number of inauthentic files transferred between peers, and was robust against collectives of malicious users comprising as much as 75% of the user population actively cooperating to attempt to subvert the system.

2 Requirements

Kamvar et al established five design principles to which P2P reputation systems should adhere [4]:

1. The system should be self-policing. The users on the system should define and enforce what is considered an acceptable level of performance for participation in the network.

2. The system should maintain anonymity. A peer's reputation should correlate with an opaque identifier rather than with an external identity.

3. The system should not assign any advantage to newcomers. Reputation should be obtained by multiple transactions at the required level of performance, and it should not be advantageous for malicious peers with poor reputations to continuously change their opaque identifiers to obtain newcomer's status.

4. The system should have minimal overhead in terms of computation, infrastructure, storage, and message complexity.

5. The system should be robust to malicious collectives of peers who know one another and attempt to collectively subvert the system.

The second requirement is a particular thorn in the side of any P2P trust management system. It is often the case that P2P file sharing systems are used to transfer documents in a questionably legal manner. As a result, many users have a high incentive to remain anonymous. The requirement of anonymity means the identity of each peer is necessarily opaque. One user could occupy many different peers on the network, each with a different apparent "trustworthiness" in the eyes of the community. This natural tension distinguishes P2P reputation systems from other reputation systems such as those mentioned above.

The first requirement that the network should be self-policing is of the most interest to us. We demonstrate that the EigenTrust system does not meet this requirement, and in doing so leaves itself vulnerable to a potentially devastating attack.

3 Overview of the EigenTrust Algorithm

At this point we will turn to a high-level description of the EigenTrust algorithm. EigenTrust uses a probabilistic trust model based on an aggregation of peer-determined local trust evaluations [4]. Each of these local evaluations of trust, c_{ij}, where the trust is from i to j,

have been normalized to values between 0.0 and 1.0. The c_{ij} values have no necessary tie to an objective performance metric, but are the subjective determination of one peer toward another, with the meanings of equal values varying between different peers in the network.

In the simple, non-distributed version of the EigenTrust algorithm, peer i asks each of its daughter peers $d \in D_i$ for their opinion c_{ij} of each of their sets of acquaintances B_d^i, from which each d has downloaded files. By continuously pushing outward in this manner and weighing the results based on how much it trusts each of its daughter peers, peer i can calculate its trust for all of the other peers on the network.

At this point i can perform the process again for each d. At this point the evaluating peer repeatedly factors in the evaluations of its second degree connections, third degree connections, and so forth, broadening the basis of its opinion of the subject peer. The evaluating peer can continue in this manner a number of steps, ultimately bounded by the number of peers in the network and the edge density of the graph, until it has factored in everyone's opinions in the network. This final value is the overall opinion of the entire community toward its member peers, and this final set of trust values is represented as a vector $\vec{t_i}$. Because of the aggregation properties of the localized value matrix, $\vec{t_i}$ will converge for every i in the network after a large enough number of iterations, so that any peer in the network would arrive at identical results if it were to perform the same series of evaluations.

It is of interest to note that this local evaluation also allows members of malicious collectives to arbitrarily give each other high trust rankings, in an attempt to hijack the ranking structure of the network and ensure that files from trustworthy peers are not recommended over files from the malicious peers.

In the distributed version of EigenTrust, an obvious issue is where to store the $\vec{c_i}$ values. In a simple system, peer i computes and stores its own $\vec{c_i}$ value, and reports this value if asked. The problems with this are obvious, however. A malicious peer could misreport its own trust value and subvert the network. In order to circumvent this, distributed EigenTrust assigns a different peer the responsibility of reporting the trust value for peer i. Of course, this just moves the problem, as a malicious score manager will report low values for other users in order to make itself seem comparatively more trustworthy.

For this reason, in the secure distributed version of EigenTrust, the $\vec{c_i}$ value for each node is calculated by numerous score managers, which are selected to represent each peer. In order to prevent individual score managers from publishing inauthentic values, majority votes are used to resolve conflicts in reporting. While this mitigates the problem described above somewhat, it clearly still does not completely resolve the challenge of guaranteeing correct reporting for any given peer i.

A summary of the secure distributed EigenTrust algorithm is presented in *Algorithm 1* below. For the purposes of understanding this algorithm, know that B_d^i represents the set of peers from which peer i has downloaded files, and that A_d^i represents the set of peers which have downloaded files from peer i [4].

Thus, the EigenTrust algorithm uses a similar model to that of PageRank [6]. PageRank treats each website as a node in a graph, and each link as a directed edge from one node to another. The algorithm establishes the value of each node in the graph by how often a random surfer would land on each node relative to all other nodes. Nodes with many in-links will often get a high value, and nodes with few in-links, but from high value sources, will also tend to get a high value. EigenTrust replaces websites with peers, and links with local

Algorithm 1 Secure Distributed EigenTrust from [4]

foreach *peer* i **do**
  Submit local trust values $\vec{c}_i$ to all score managers at positions $h_m(pos_i)$, m=1...M-1;
  Collect local trust values $\vec{c}_d$ and sets of acquaintances B_d^i of daughter peers $d \in D_i$;
  Submit daughter d's local trust values c_{dj} to score mgrs $h_m(pos_d)$, m=1...M-1, $\forall j \in B_d^i$;
  Collect acquaintances A_d^i of daughter peers;
  foreach *daughter peer* $d \in D_i$ **do**
    Query all peers $j \in A_d^i$ for $c_{jd}p_j$;
    repeat
      Compute $t_d^{(k+1)} = (1 - a)(c_{1d}t_1^{(k)} + c_{2d}t_2^{(k)} + ... + c_{nd}t_n^{(k)}) + ap_d$;
      Send $c_{dj}t_d^{(k+1)}$ to all peers $j \in B_d^i$;
      Wait for all peers $j \in A_d^i$ to return $c_{jd}t_j^{(k+1)}$;
    until $|t_d^{(k+1)} - t_d^{(k)}| < \epsilon$;
  end
end

trust, but is otherwise identical in its basic approach.

A naive implementation of this algorithm makes it easy for the random surfer to get stuck inside a closed loop in the graph (the rank sink problem[6]). The solution is to add a small probability that instead of jumping to one of the nodes pointed to by the current node, as is normally the case, the surfer jumps elsewhere, usually with about a 15% probability. Whereas PageRank defines this "elsewhere" as a uniform distribution over all nodes in the graph, EigenTrust introduces the notion of a pre-trusted peer.

4 Pre-trusted Peers

Both the distributed and the simple versions of EigenTrust require that each node i has at least one daughter peer or acquaintance in the network to begin with. Without a foothold in the network, newcomers are unable to get reliable trust information. To circumvent this issue, the literature on the EigenTrust algorithm suggests providing "*a priori* notions of trust" in the form of "pre-trusted peers" [4]. When a new peer joins the network and does not yet know anyone, the peer uses the perspective of the network provided by the pre-trusted peers, from whom it can learn who else to trust. As a result of their high degree of connectedness in the network, pre-trusted peers enjoy significantly higher trust values than other peers in the network, and are instrumental in shaping how the network views trust. It is the contention of this paper that this practice not only undermines the first design goal of a P2P reputation system, but that it exposes the entire network to extraordinary degrees of risk.

A key assumption in the decision to include pre-trusted peers was that "the first few peers to join a network are often known to be trustworthy" [4]. In part, the assumption is weak because it expects a trustworthy peer to remain so throughout the course of its membership in the P2P network. While we might reasonably argue as much about the users of a system,

the integrity and trustworthiness of their computers is quite another matter. Consider the possibility of a computer worm that transmits itself over a P2P file sharing network, or the possibility that pre-trusted peers could be compromised legally by organizations that do not want the P2P network to function.

The concept of pre-trusted peers undermines the first design consideration of P2P reputation systems, which is to provide a self-policing system. If the globally aggregated trust rankings are heavily informed by an arbitrarily determined initial set of pre-trusted peers, then it becomes possible to subvert the global ranking mechanism by compromising one of the pre-trusted peers. This may even be done unbeknownst to the owner of that peer. By reporting high degrees of confidence in known malicious peers, this malicious pre-trusted peer can spread a distorted view of the network to users, weakening trust among all worthy peers, and perhaps more importantly, weakening the confidence that users have in the system.

5 Federated EigenTrust

As a result, getting rid of pre-trusted peers entirely is desirable. However, something must take their place as they are necessary components of the computation of trust previously described. Pre-trusted peers are also responsible for the outstanding performance that EigenTrust achieves. Networks with a majority of malicious nodes could not perform so well without the pre-trusted peers identifying the good minority. For this reason, we preserve the general concept of pre-trusted peers, but actually replace them with regular nodes that represent the trust of a section of the network.

5.1 Representative Nodes

We introduce the idea of "representative nodes" to take the place of pre-trusted peers in the trust computation. The power of pre-trusted peers came from the users behind them who participate normally in the network and judge the quality of downloads. Federated EigenTrust utilizes this same concept.

The idea is that each regular node in the network is capable of becoming a representative of another node based on the trust they have for each other. These two nodes will "fold" together and one of them will be elected the representative node of the other. This representation is used to eliminate malicious collectives by never allowing regular nodes to elect a malicious representative.

Algorithm 2 works as follows. Each peer in the network that does not have a representative looks at its trust relationship with every other peer in the network. If a positive trust relationship exists, and the other peer does not have a representative, the peer is noted as a candidate for the upcoming "mini election." Once all candidates have been found, the peer selects one of them to form a partnership with and a "mini election" occurs. The "mini election" consists of the partners randomly selecting a representative between them. The peer chosen as representative will be able to continue "folding" with future peers, the other will not. This representative will be also be among the candidates for pre-trusted peer replacement later.

Algorithm 2 Representative Formation

foreach *peer* i **do if** i does not have a representative
  foreach other *peer* j **do**
    if j does not have a representative and i,j have mutual trust
      j is a candidate for representation with i
    end
  end
  select a random candidate c
  either i represents c or c represents i by fair coin toss
end

5.2 Personalization

Our solution is to make use of personalized EigenTrust [2]. In the original EigenTrust, the pre-trusted peers were identical for the entire network. Personalized EigenTrust is a very simple modification that allows each node in the network to take a different perspective on who the pre-trusted peers are in the network. In Federated EigenTrust, each peer in the network will choose some subset of the representative nodes to consider its pre-trusted peers. This way, sections of the network that have provided malicious files can be excluded from being potential personal pre-trusted peers.

Algorithm 3 Personalized Peer Selection

form *trusted peers list* from top level representatives
foreach *peer* i **do**
  sort the *trusted peers list* by number of bad downloads i has received
  select best x remaining peers from the personalized list where x is chosen by client
end

Algorithm 3 selects the representative nodes that each peer will consider pre-trusted peers. A trusted peer list is formed from the globally available top level representatives. Then each peer will personalize this list by removing from consideration any member that has provided large numbers of bad downloads. Then the peer will select the best x remaining peers from the personalized trusted peers list. This x value is selected by the client.

The downside to this approach is that a new user to the network will not get a very good picture of the trustworthiness of other peers, because they will not have trust in anyone and will not be able to reliably choose a representative. This method also introduces additional complexity to the computation of trust values that now need to be computed on a personalized individual basis for each peer, but efficient methods have been found to compute these personalized trust values[3].

5.3 Distributed Implementation

In order to distribute *Algorithm 2*, instead of each peer iterating through each other peer in the network, it would select a candidate at random for which positive trust exists. The node can query the candidate to see if the trust is mutual and if so, the node will attempt to "fold" with the candidate.

The distribution of *Algorithm 3* would require a secure distributed disjoint set data structure to store the nodes that are top level representatives. Assuming we have such a data structure, each node can perform evaluation of network subsections by attributing bad downloads to the representative of that peer. At this point, we would apply Personalized EigenTrust normally.

6 Simulation Design

For our simulation we implemented the P2P file sharing network as a graph, composed of nodes representing each peer and edges representing trust relationships between the nodes. Each node can be flagged as benign or actively malicious, and for each case, a certain percentage of the node's responses for file transfer will be an inauthentic file. In the case of malicious nodes, inauthentic files made up 100% of responses, whereas in non-malicious nodes, inauthentic files were present in only 5% of responses.

The simulation instantiates the network graph, populates it with peers according to its parameters, and then walks the network through a series of steps in time, during which each node may attempt to conduct a file transfer with another node. In order to do so, each node will attempt to determine the most trustworthy peers with whom to carry on the transaction. To avoid the situation where downloads occur only from the pre-trusted peers to the exclusion of the rest of the network, a small fixed probability of downloading from a peer with no trust is added.

Our simulation is very optimistic in its treatment of real adversaries on the network. We do not consider the actual topology of the network as more sophisticated models do[7], and instead assume all peers are aware of each other. We also provide no way for a non-malicious peer to be transformed into a malicious peer, or *vice versa*, and assume *perfect* analysis of malicious downloads so that peers always know if they have received an inauthentic file. Finally, we only consider a single threat model: that of a single malicious collective that always trusts itself and no one else. We took this coarse approach to the design of the simulation in order to more effectively isolate and analyze the results of including malicious pre-trusted peers in the network.

7 Results

Despite the simplified model of our simulation, our results agree well with those of the EigenTrust paper when running standard EigenTrust. The network is surprisingly capable of maintaining high quality results even in the face of an overwhelming malicious collective.

In addition, our hypothesis that the pre-trusted peers are a major weakness of EigenTrust is confirmed. In fact, a single compromised pre-trusted peer is capable of subverting the trust

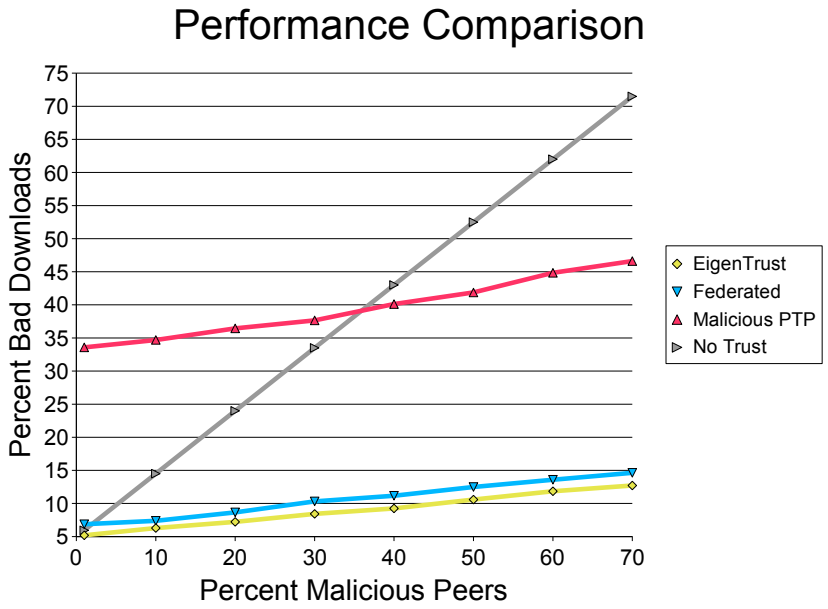

Figure 1: Performance of the network when all the pre-trusted peers are trustworthy using standard EigenTrust, when one of the pre-trusted peers is compromised in standard EigenTrust by a malicious peer, when pre-trusted peers are based on representative nodes, and when there is no trust metric at all

system so thoroughly that it performs worse than with no trust system at all, dramatically so at realistic levels of malicious peers (below 20%).

8 Analysis

8.1 Adherence to the Five Design Principles

Our P2P reputation system addresses each of the five design principles as well or better than traditional EigenTrust.

1. The system is self-policing. In traditional EigenTrust, the definition of trustworthiness is set by the global "pre-trusted peers" and individual users have little influence over that definition. In our system, individual users in the system make their own choices as to the subset of the networks they consider trustworthy.

2. The system maintains anonymity. In this respect, our system does not differ from traditional EigenTrust. Users are opaque identifiers that award trust and evaluate trustworthiness to other opaque identifiers.

3. Neither traditional EigenTrust, nor our system completely adhere to the principle by which a newcomer gains no advantages. Both a completely distrusted peer and a newcomer share a perceived trust value of 0.0. However, a completely distrusted peer must have a history of negative transactions that must be overcome to achieve positive

trust, whereas a newcomer would begin to achieve positive trust immediately with the first positive transaction. To that end, a newcomer has a lesser disadvantage than a known malicious peer. With the anonymity design principle, it is impossible to perfectly address this issue.

4. The system has minimal overhead in terms of computation, infrastructure, and message complexity. Traditional EigenTrust can be computed securely in a low overhead distributed fashion[4]. Our system, as described, performs trust computation with the same complexity as EigenTrust, but contains an additional distributable peer fold operation, and a cluster tree mapping algorithm that can be implemented as an efficient distributed disjoint set data-structure.

5. The system is robust against malicious collectives of peers who know one another and attempt to collectively subvert the system. In order to achieve substantial influence in the network, a malicious collective would need to fold into a large representative subsection of the network. This large representative subsection provides a single target for identification and elimination by users. Malicious nodes could attempt to form multiple smaller representative subsections of the network, but these subsections would be less attractive to users than large non-malicious subsections.

8.2 Potential Attacks

A potential vulnerability in our system is the method by which nodes are grouped, or folded, under a representative. In our current grouping implementation, nodes are only concerned with mutual trust at the time of a fold between two nodes. This leaves the network vulnerable to an attack by initially well behaving nodes that are designed to behave maliciously once they are incorporated into a representation group. Since nodes enter groups by folding with the root, malicious nodes incorporated into a group can corrupt the group without impacting the viability of the fold from the perspectives of new nodes.

Another potential vulnerabilities lies in the manner by which individual nodes select subsection viability. In our implementation, individual nodes considered a network subsection to be viable if the number of malicious downloads received from its members was less than twice of the subsection that delivered fewest malicious downloads. If a malicious collective compromised each of the major network subsections to the same degree, all of the subsections could still be considered viable, even if the rate of bad downloads from the network subsections increased.

Both of these potential vulnerabilities can be partially addressed by implementing a turnover rate for the group relationships. If nodes left and joined network subsections, nodes that maintained high quality trust relationships with other nodes would maintain their placement within a subsection. However, nodes that began to exhibit malicious behavior would not be in a position of mutual trust necessary to rejoin a subsection.

8.3 Avenues of Future Research

A major avenue of future research would be in producing a more effective method of network partitioning into subsections. The tree-forming representative fold method implemented in

our simulation was successful in defending against malicious nodes continually offering bad downloads. However, this method is vulnerable to a transitivity attack. A malicious peer may provide good files to some nodes, and bad files to other nodes, while trusting those it provides good files to enough to "fold" with them. As a result, a malicious peer could be folded into a good part of the network, since it would have mutual trust with a benign node.

Additionally, realistic networks need to function despite the continual addition and subtraction of nodes from the network. Concrete criteria for breaking and reforming representative relationships between nodes, as a part of partitioning, would be essential before this system could be implemented outside of a simulated network.

9 Conclusion

The inclusion of even a single member of a malicious collective into the pre-trusted peer group of the standard EigenTrust P2P reputation system can have severe negative consequences on the reliability of the reported trust rankings for peers in the network. At practical sizes for malicious collectives, it results in performance significantly worse that having no trust system at all. At larger sizes, even though a compromised pre-trusted peer performs better than no trust system, it is already such a poor quality network that it would be unusable.

For this reason, we believe the inclusion of pre-trusted peers in the EigenTrust reputation management system is a potentially serious flaw that results from weak adherence to the first design consideration for P2P reputation systems outlined in [4]–that the system should be self-policing. The pre-trusted peers impose upon the rest of the network in such a significant way that the network could hardly be called self policing.

We believe we have demonstrated that the EigenTrust system can function without preselected pre-trusted peers, without significant detriment. Our results show that forcing peers to experiment to determine for themselves which parts of the network to trust does not result significant degradation of the quality of downloads. Peers are still able to find good files under impressive saturation of the network by a malicious collective, and without the vulnerability that introducing pre-trusted peers exposes the network to. However, we also point out a number of flaws that our system presents that would have to be solved before it would be possible to use on a real network.

10 References

[1] G.A. Akerlof. The Market for Lemons: Quality Uncertainty and the Market Mechanism. *Readings in Social Welfare: Theory and Policy*, 2000.

[2] P.A. Chirita, W. Nejdl, M. Schlosser, and O. Scurtu. Personalized reputation management in P2P networks. In *Proceedings of the ISWC 2004 Workshop on Trust, Security, and Reputation on the Semantic Web, Hiroshima, Japan*, volume 7, 2004.

[3] G. Jeh and J. Widom. Scaling personalized web search. In *Proceedings of the 12th international conference on World Wide Web*, pages 271–279. ACM Press New York, NY, USA, 2003.

[4] Sepandar D. Kamvar, Mario T. Schlosser, and Hector Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM.

[5] R.A. Malaga. Web-Based Reputation Management Systems: Problems and Suggested Solutions. *Electronic Commerce Research*, 1(4):403–417, 2001.

[6] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web, 1998.

[7] A. Schlosser, M. Voss, and L. Bruckner. On the simulation of global reputation systems. *Journal of Artificial Societies and Social Simulation*, 9(1):1, 2006.

[8] L. Xiong and L. Liu. PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, pages 843–857, 2004.

[9] R. Zhou and K. Hwang. PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing. *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, pages 460–473, 2007.