New Metrics for Reputation Management in P2P Networks

D. Donato[1], M. Paniccia[2], M. Selis[2],
C. Castillo[1], G. Cortese[3], S. Leonardi[2]

1. Yahoo!Research Barcelona – Catalunya, Spain
2. Sapienza University of Rome – Rome, Italy
3. Consorzio Università Industria, Radiolabs, University of Rome, Tor Vergata – Rome, Italy

P2P networks and reputation

Preliminaries

Threat models

Metrics

Evaluation

P2P networks features

- ✓ Resource sharing: bandwidth, storage space, and computing power
- ✓ Information sharing
- ✓ Lack of central authority
- ✗ Lack of guarantee and certification of the shared resources

Downside

The open and anonymous nature of P2P networks opens doors to manipulation of the services (information) provided

Downside

The open and anonymous nature of P2P networks opens doors to manipulation of the services (information) provided

The open and anonymous nature of P2P networks makes it difficult to calculate reliable quality metrics for peers and objects

Reputation management

Reputation management is used to:

- Describe the performance of peers in the network
- Describe how reliable they are

Reputation management

Reputation management is used to:

- Describe the performance of peers in the network
- Describe how reliable they are

Such mechanisms should be robust against malicious peers.

Starting point

EigenTrust

We start with EigenTrust [Kamvar, Schlosser and Garcia-Molina, 2003], an algorithm designed for reputation management in file sharing application over p2p networks.
We combine EigenTrust with metrics of reputation computed using techniques recently introduced for detecting and demoting Web Spam.

Contribution

- We integrate Truncated PageRank [Becchetti et al., 2006], Estimation of Supporters [Palmer et al., 2002] and BadRank in reputation management
- We introduce a number of new threat models
- We test existing and new threat models in a simulated environment
- We show that our combined approaches perform better than EigenTrust alone in reducing the amount of inauthentic downloads

EigenTrust

Definition of local trust in EigenTrust

We define a local trust value s_{ij} as

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

In order to avoid malicious peers to assign arbitrarily high local trust values, it is necessary to normalize them. The normalized local trust value is c_{ij} is defined as follows:

$$cij = \frac{max(s_{ij},0)}{\sum_j max(s_{ij},0)}.$$

EigenTrust

Hypothesis

Peers who are honest about the files they provide are also likely to be honest in reporting their local trust values.

EigenTrust

Global trust

The idea of transitive trust, inspired by PageRank [Page et al., 1998], leads to a system where trust values propagate through paths along the network

PageRank

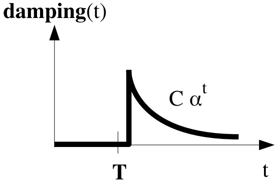
PageRank can be expressed as a weighted summation of paths of varying lengths

$$S = \sum_{t=0}^{\infty} \frac{\text{damping}(t)}{N} P^t \ .$$

t: the lengths of the paths.
damping(t): decreasing function of t.
P: row-normalized citation matrix

Truncated PageRank

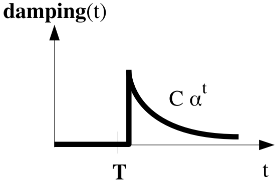
Proposed in [Becchetti et al., 2006]. Idea: reduce the direct contribution of the first levels of links:



$$\text{damping}(t) = \begin{cases} 0 & t \leq T \\ C\alpha^t & t > T \end{cases}$$

Truncated PageRank

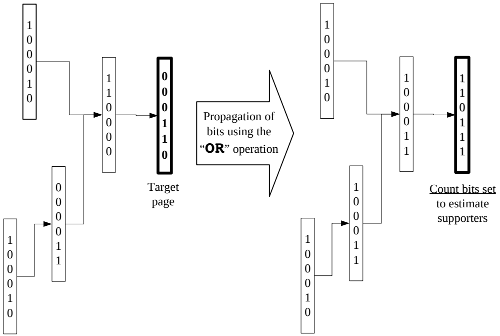
Proposed in [Becchetti et al., 2006]. Idea: reduce the direct contribution of the first levels of links:



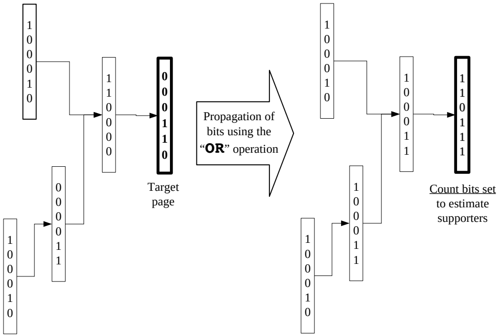
$$\text{damping}(t) = \begin{cases} 0 & t \leq T \\ C\alpha^t & t > T \end{cases}$$

☑ No extra reading of the graph after PageRank

Estimation of supporters



Estimation of supporters



[Becchetti et al., 2006] shows an improvement of ANF algorithm [Palmer et al., 2002] based on probabilistic counting [Flajolet and Martin, 1985]. After d iterations, the bit vector associated to any page x provides information about the number of supporters of x at distance $\leq d$.

This algorithm can be used to estimate the number of different peers contributing to the ranking of a given peer.

BadRank

If a page links to another page with a high BadRank, then also this page should be considered a page with negative characteristics. The difference with respect to PageRank is that BadRank is not based on the evaluation of inbound links of a web page but on its outbound links.

$$br(i) = d \sum_{i \to j} \frac{br(j)}{indeg(j)} + (1-d)e(i)$$

computed on the graph of negative evaluations

Network Models

Transaction Network

A link from a node (peer) i to a node j is inserted every time i downloads a file from j. Each link is weighted with a positive value if the downloaded file was authentic, negative otherwise.

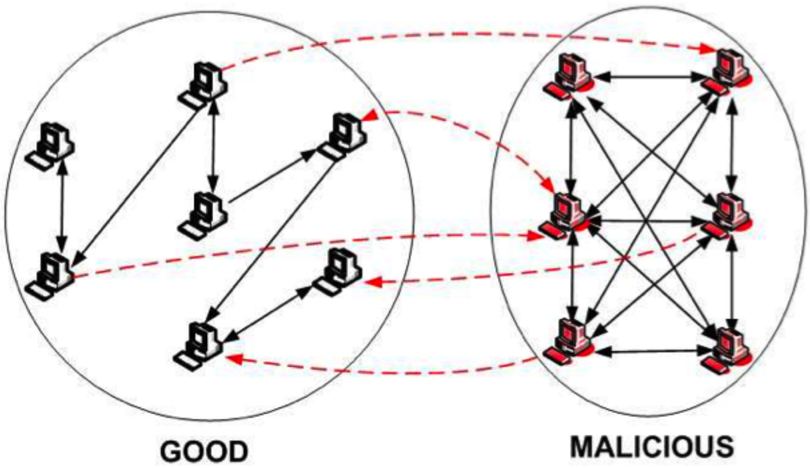
Positive Opinion Network

A link is inserted from a node i to a node j only after the download of authentic files.

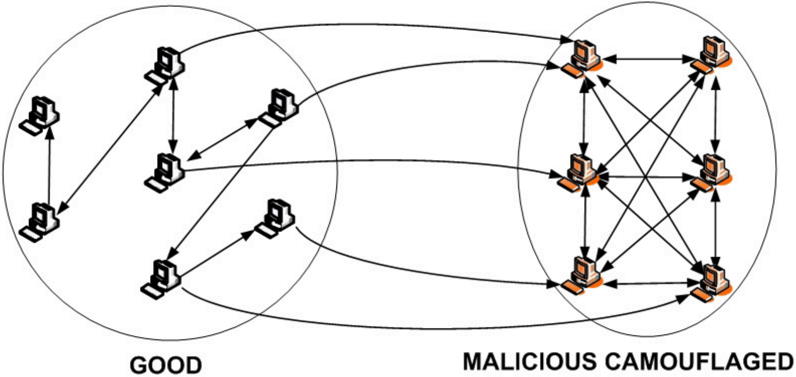
Inverse Network

The transpose of the positive opinion network.

Threat Model A (individuals) and B (collective)

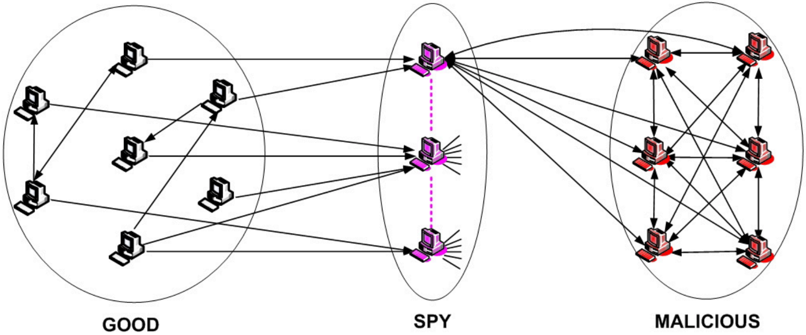


Threat Model C - collectives with camouflage



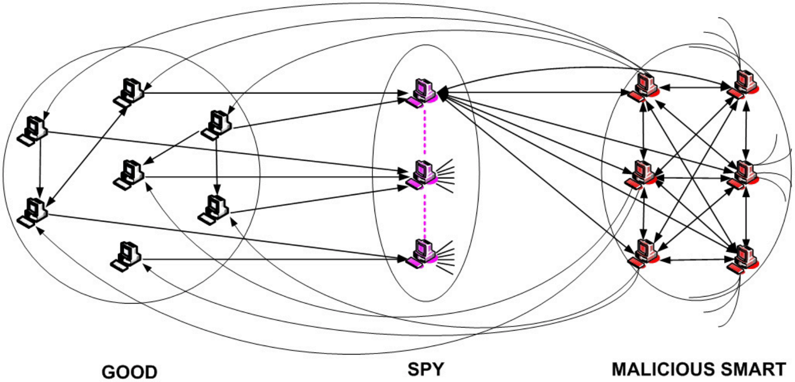
They provide good files sometimes

Threat Model D



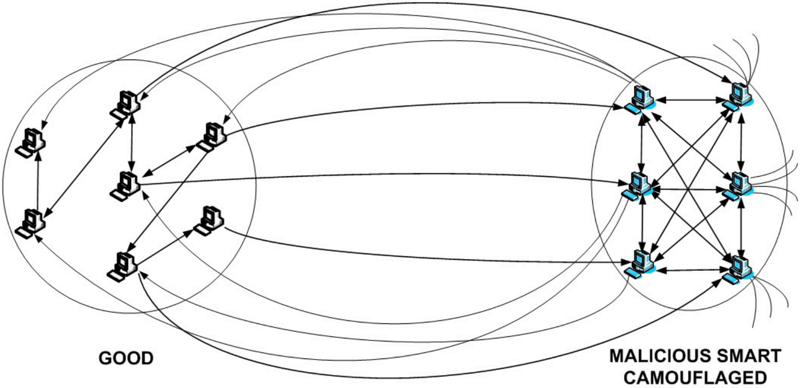
Have a set of nodes providing good ratings for them

Threat Model G - malicious smart model



Sometimes give ratings to the rest of the network

Threat Model H - malicious smart model with camouflage



Sometimes provide authentic files and ratings to the rest of the network

Eigentrust with Inverse Eigentrust - Model D

Encourage peers to provide ratings about other peers

Require: EigenTrust score vector ET, Inverse EigenTrust score vector I

```
if I[i] > 0 then
    return ET[i]
else
    return 0
end if
```

Eigentrust with Inverse Eigentrust - Model G

Encourage peers to provide many ratings about other peers

Require: EigenTrust score vector ET, Inverse EigenTrust score vector I, threshold $tr = \sum_i \frac{ET[i]}{N}$

```
if I[i] ≥ tr then
    return ET[i]
else
    return 0
end if
```

EigenTrust with Truncated PageRank

Malicious peers receive positive values from the other members of the coalition (malicious and spy). This means that the most of the *trust mass* is propagated starting from nodes at few hops of distance.

Require: Eigentrust score vector ET, Truncated PageRank vector P, threshold tr

if $P[i] \geq tr$ **then**
    return $ET[i]$
else
    return 0
end if

EigenTrust with Estimation of Supporters

Malicious peers supporters necessarily belong to the same coalition. This means that a malicious peer obtain an high reputation because of the great number of supporters at short distance from it.
The Bit Propagation algorithm can be used to perform an analysis of the connectivity of the transition network in order to detect local anomalies.

Require: EigenTrust score vector ET, Bit Propagation vector BP, threshold tr

if $BP[i] \geq tr$ **then**
  return $ET[i]$
else
  return 0
end if

Badness

Propagating badness

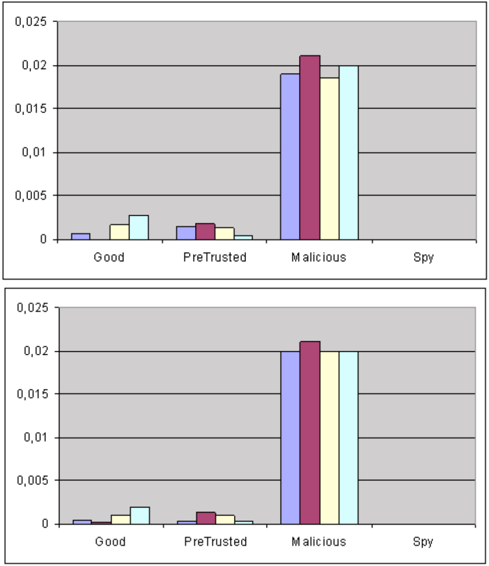
If i trusts j and j distrusts k then, with high probability, also i should regard k as untrustworthy. We can define the **Global Badness** as:

$$\mathbf{negT} = \mathrm{D}^{\top}\mathbf{T}$$

where D is the normalized negative opinion matrix and $\mathbf{T}$ is the EigenTrust Rank. Each peer i has a global Badness given by

$$\mathbf{negT_i} = \sum_{j=1}^{n} negC_{ji} \times \mathbf{T_j}$$

Average BadRank for models A-D



Average BadRank after 25 and 50 cycles.

Dishonesty

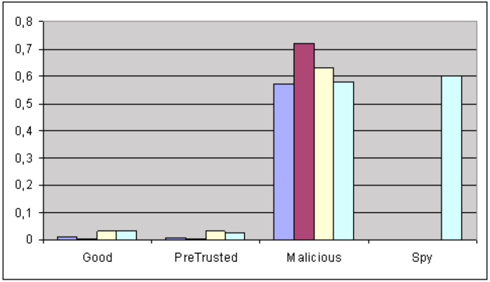
The badness is able to differentiate between good and malicious peers but it does not help in discovering spies.

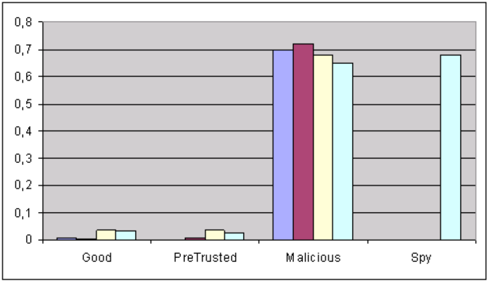
We measure **dishonesty**:

$$\mathbf{dishonesty_i} = \sum_{j \in P} \mathbf{negT_j}$$

where P is the set of peers that i have given positive ratings
The dishonesty is high for all those peers which give good ratings to peers with high badness.

Average dishonesty for models A-D





Average Dishonesty after 25 and 50 cycles.

Settings

- 100 good peers
- 5 pre-trusted peers
- probability to supply corrupted files equals to 2% for good peers

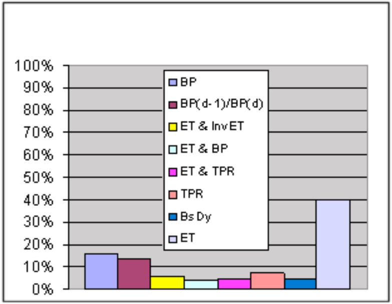
Settings

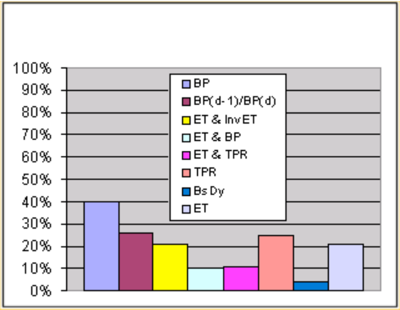
- 100 good peers
- 5 pre-trusted peers
- probability to supply corrupted files equals to 2% for good peers

Evaluation

We consider the average ratio between the number of inauthentic downloads and the total number of downloads

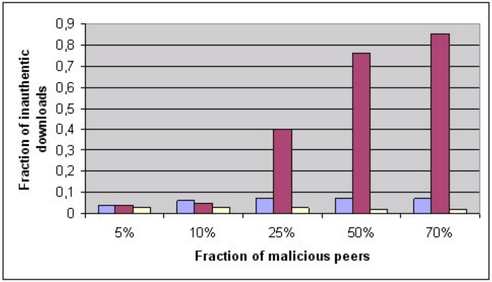
Comparison

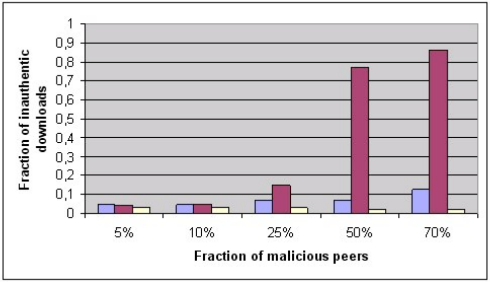




Inauthentic downloads for threat model D (malicious and spies) and threat model G (plus smartness)

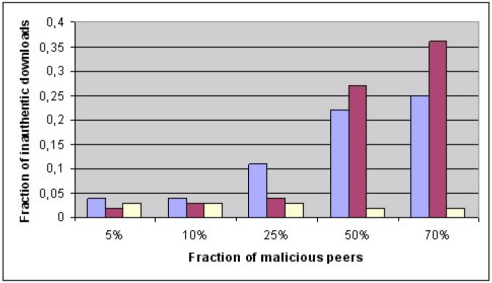
Threat models A (individuals) and B (collective)

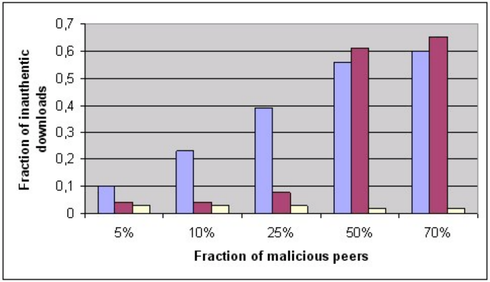




EigenTrust, E. + TruncatedPR, E. + badness + dishonesty

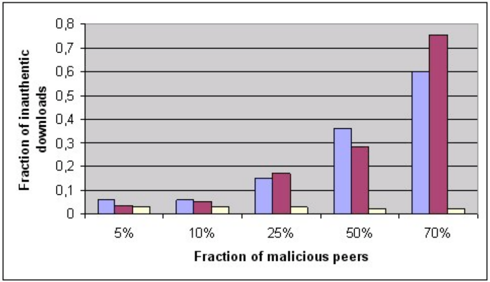
Threat model C (camouflage) and D (spies)

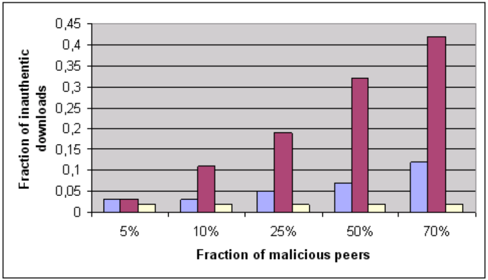




EigenTrust, E. + TruncatedPR, E. + badness + dishonesty

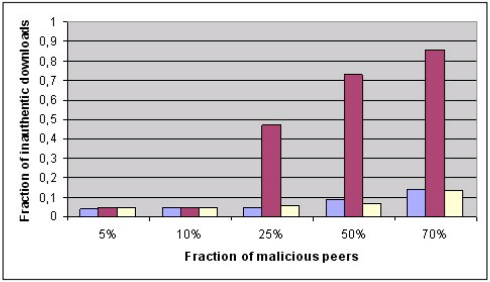
Threat model G (smart) and H (smart+camouflage)

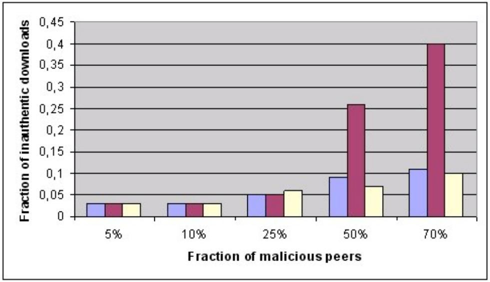




EigenTrust, E. + TruncatedPR, E. + badness + dishonesty

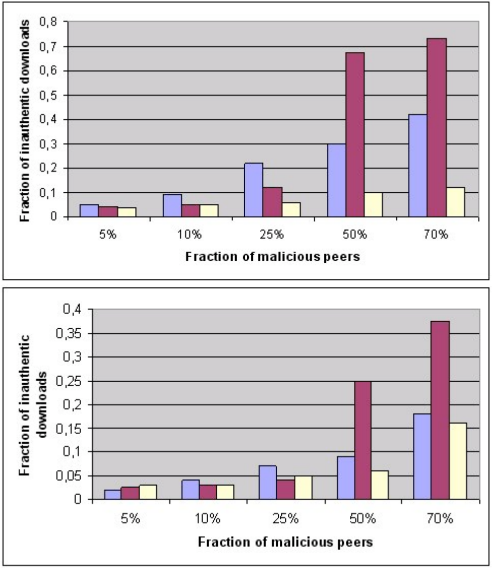
Variant: provide bad files, but be honest





Threat model A',C'

Variant: provide bad files, but be honest; combined attacks



Threat model D+A',D+C'

EigenTrust vs. EigenTrust + Badness and Dishonesty

Malicious	A	B	C	D
5%	4%/3%	4%/3%	4%/3%	10%/3%
10%	6%/3%	6%/3%	4%/3%	23%/3%
25%	7%/3%	7%/3%	11%/3%	39%/3%
50%	8,5%/2%	10%/2%	22%/2%	56%/2%
70%	14%/2%	15.5%/2%	25%/2%	60%/2%

A'	C'	D+A'	D+C'	G	H
4%/4%	3%/3%	5%/4%	3%/3%	6%/3%	3%/2%
5%/5%	3%/3%	9%/5%	4%/3%	6%/3%	3%/2%
5%/5%	6%/5%	22%/6%	7%/5%	15%/3%	5%/2%
8%/7%	9%/7%	30%/8%	9%/6%	36%/2%	7%/2%
13%/11%	11%/10%	42%/12%	18%/13%	50%/2%	12%/2%

Comparison on all attacks

Precision vs recall

Set threshold for identifying malicious peers:

- Recall: % malicious peers identified
- Precision: number of false positive

$$T'_j = \begin{cases} 0 & \text{se } badness_j > BadnessThreshold \\ T_j & \text{otherwise} \end{cases}$$

$$sel_j^{Badness} = \begin{cases} sel_j^{Eig} & \text{se } \frac{average\ \ badness}{average\ \ global\ \ trust} < r_{bad} \\ 90\% * \frac{T'_j}{\sum_{i=1}^{R} T'_i} & \text{se } T'_j > 0\ and\ \frac{average\ \ badness}{average\ \ global\ \ trust} >= r_{bad} \\ 10\% & \text{se } T'_j = 0\ and\ \frac{average\ \ badness}{average\ \ global\ \ trust} >= r_{bad} \end{cases}$$

Results:

- 70% recall
- less than 2% false positive

What's next

- We have discussed several threat models and tools

What's next

- We have discussed several threat models and tools
- Find more general threat models (not easy!)

What's next

- We have discussed several threat models and tools
- Find more general threat models (not easy!)
- Better integration of existing tools

What's next

- We have discussed several threat models and tools
- Find more general threat models (not easy!)
- Better integration of existing tools
- Propose more tools that increase the cost of attacks and/or make them less successful

What's next

- We have discussed several threat models and tools
- Find more general threat models (not easy!)
- Better integration of existing tools
- Propose more tools that increase the cost of attacks and/or make them less successful
- Propose techniques that can adapt to different environments (e.g.: learn how hostile is the network currently, behave accordingly)

What's next

- We have discussed several threat models and tools
- Find more general threat models (not easy!)
- Better integration of existing tools
- Propose more tools that increase the cost of attacks and/or make them less successful
- Propose techniques that can adapt to different environments (e.g.: learn how hostile is the network currently, behave accordingly)

What's next

- We have discussed several threat models and tools
- Find more general threat models (not easy!)
- Better integration of existing tools
- Propose more tools that increase the cost of attacks and/or make them less successful
- Propose techniques that can adapt to different environments (e.g.: learn how hostile is the network currently, behave accordingly)

Thank you!

http://ewwws.com/pr/przero.php.

PR0 - Google's PageRank 0 Penalty.

Becchetti, L., Castillo, C., Donato, D., Leonardi, S., and Baeza-Yates, R. (2006).

Using rank propagation and probabilistic counting for link-based spam detection.

In *Proceedings of the Workshop on Web Mining and Web Usage Analysis (WebKDD)*, Pennsylvania, USA. ACM Press.

Flajolet, P. and Martin, N. G. (1985).

Probabilistic counting algorithms for data base applications.

Journal of Computer and System Sciences, 31(2):182–209.

Kamvar, S. D., Schlosser, M. T., and Garcia-Molina, H. (2003).

The eigentrust algorithm for reputation management in p2p networks.

In *WWW*, pages 640–651.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1998).
The PageRank citation ranking: bringing order to the Web.
Technical report, Stanford Digital Library Technologies Project.

Palmer, C. R., Gibbons, P. B., and Faloutsos, C. (2002).
ANF: a fast and scalable tool for data mining in massive graphs.
In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 81–90, New York, NY, USA. ACM Press.