Detecting Deceptive and Malicious Voting Behaviors in Decentralized Systems

Eric Chang

Yale University

Abstract. Deceptive behaviors of peers in today's decentralized systems have become a serious problem due to the anonymous and self-organization nature. In this paper, we propose Soc, a novel *active challenge-response mechanism* based on the notion that the one side of transaction with preponderant knowledge can detect whether the other side is telling a lie. In Soc, through introducing the friend-based scheme, each peer can establish own friend relationships quickly. With the secret information of friends, Soc can construct the *asymmetrical information* between peers. Our active challenge-response mechanism can help peers find the deceivers in system based on the asymmetrical information. Soc also provides the mechanism which can reduce the probability of impact brought by deceptive peers. Compared with existing reputation models, Soc is more robust to the problems of *collusive deceivers* and *cold start*. The evaluation results illustrate that Soc can effectively address the problem of deceptive peers.

1 Introduction

Decentralized applications, e.g., BitTorrent, eDonkey, Gnutella, KaZaA, have become increasing popular. However, due to the anonymous and self-organization nature, the participants of P2P network have to face some potential risks involved in the application transactions without adequate experience and knowledge about other participants [1], [2], [3], [4]. Many studies indicated that P2P networks are highly vulnerable to deceptive peers. For a typical deceptive behavior, individual or collusive deceptive peers provide fake or misleading feedback during the transaction between other peers to achieve their malicious purposes [5] and [6].

Previous studies on addressing deceptive peers mainly focus on the reputation model. Existing reputation models such as EigenTrust [7], PeerTrust [6] and Scrubber [8] identify the deceptive peer by calculating a reputation score for each peer. Credence [9], an object reputation model, calculates the reputation for each file based on the experiences between peers [10], [11], [12], [13], [14], [15], [16], [17], [18], [19], [20].

Due to the passive features of reputation models, the peers in above systems are always in defense position and they can only resist the deceptive peers with the mechanisms provided by the models. For those deceptive peers collude in cheating to raise their reputations, the current reputation models can not give a good solution. Moreover, for a new peer joins in the system, the existing reputation models can not provide a reasonable initial reputation value for this new member [5]. Due to lack of enough transaction, a new peer is easy to be deceived. We call this problem as *cold start*.

In this paper, to address the deceptive behaviors of peers in P2P networks, we introduce Soc, a novel active challenge-response mechanism. Soc encompasses three key techniques to detect and punish those deceptive peers. *First*, through introducing a friend-based scheme, Soc can help each peer establish the friend relationships in P2P networks. In Soc, friends can share own knowledge with each other; however, friends' information is secret for others. With the secret information of friends, we construct the *asymmetrical information* between both sides of transaction. *Second*, Soc uses *active challenge-response mechanism* to help peers find the deceivers in system based on asymmetrical information. In the mechanism, we can make use of the *coincident knowledge* of both our friends and the respondent of challenge-response to detect whether the latter is a deceiver or not. And *third*, we also provide the mechanism which can reduce the probability of impact brought by deceptive peers to achieve the purpose of punishing deceivers.

The fundamental insight driving our works is that, in one round of interaction, the one side of transaction with preponderant knowledge can detect whether the other side is telling a lie. To the best of our knowledge, none of previous work focused on using active approach to address the problem of deceptive peers in P2P networks. Our research contributions are as follows:

- With the friend-based scheme, peers can obtain the knowledge quickly via either adding the friends in reality or joining some *friend-communities*. Thus, Soc can address the problem of cold start.
- Using asymmetrical information, Soc can detect the deceptive peers in networks. Soc can be robust to the collusive deceptive peers.
- With penalty mechanism, Soc can reduce the probability of impact brought by deceptive peers.

The rest of this paper is organized as follows. Section 2 will analyze both the problems of deceptive peers and current reputation models. Section 3 will present the details of Soc. Our experiments and evaluations will be discussed in Section 4. In Section 5, we will discuss conclusion and future work.

2 Analysis

Due to the anonymous and self-organization nature, the problem of deceptive participants in P2P networks is always serious. To address the problem, reputation models are introduced to P2P networks. Current reputation models indeed solve the problem of individual deceiver; however, they can not completely defend against the collusive deceivers. Moreover, the cold start problem is also not paid much attention by existing reputation models.

For the current P2P reputation models, the collusive deceivers can occupy the preponderant position in most transactions. Each time when we send a request, they are very clear about the purpose of our querying and then return an aforethought feedback; however, we have no idea about the role they play in the networks. Thus, for the passive mechanism, it is difficult to find out those deceptive peers effectively. To sit on the preponderant position, we need to construct asymmetrical information in symmetrical P2P networks and detect those deceivers with the asymmetrical information actively.

3 Soc Design

Before discussing details, we will describe the infrastructure of Soc. Soc introduces friend-based scheme to P2P networks, and for some well-known peers, e.g., ultra peers, they can maintain some friend-communities. Thus, the peers who have the same interests can establish friend relationships easily. In Soc, each peer needs to maintain two lists: the one is *friend list* which contains the information of friends, and another one is *respondent list* which is comprised of the peers who have ever been *challenged* by the list owner. Note that, the friend list of a peer can not be seen by others. Using secret friend information, Soc can construct the asymmetrical information between both sides of challenge-response.

The details of Soc will be discussed in this section. In Section 3.1, we will describe the friend-based scheme and how Soc constructs the asymmetrical information. In Section 3.2, both the mechanisms of challenge-response and penalty of Soc will be discussed. Section 3.3 will discuss the practical issues of Soc. We will present the summary of Soc in Section 3.4.

3.1 Friend-based Scheme

Soc introduces the friend-based scheme to P2P networks. Here, we present the details of scheme.

Friend List: In Soc, each peer has many friends and saves their information in friend list. Besides necessary identifier of friend, the friend list of a peer stores friends' knowledge. For instance in Fig. 1, *Alice* has two friends and their knowledge is open to *Alice*. In this instance, friends' knowledge is the evaluation of file. *Alice* will timely update the information of friend list with the new knowledge from her friends. Note that, in Soc, the friend list will only be seen by the peer himself and other peers can not see it.

Establishment of Friend Relationships: Any peer can be invited by an existing peer in the network and thus added into system; meanwhile the latter will become the friend of the new peer automatically. For Soc, friends can be the real-world acquaintances, or the online friends extracted from *MSN* or *Facebook* who are much more trustworthy than other common peers online. Soc also provides the mechanisms of friend-community and adding friend manually.

In Soc, friend relationship is symmetric, a peer needs to send a request to the other peer for adding himself as a friend and then the friend relationship can be established after the other side's agreement.

Constructing Asymmetrical Information: Because the friend list will only be seen by the peer himself, Soc can construct asymmetrical information in symmetrical P2P networks with the secret friend list. Fig. 1 presents the instance for the asymmetrical information between two peers. As shown in Fig. 1, *Bob* can only know that *Alice* has *File6*, however, *Alice* also owns the knowledge from both of *friend1* and *friend2*, for instance, *File2* is polluted. If *Alice* wants to detect whether *Bob* is a deceiver, she can ask him both the evaluations of *File2* and *File5*.

Effectiveness: Soc can address the cold start problem using the friend-based scheme and the asymmetrical information. When a new peer joins in the system, the existing

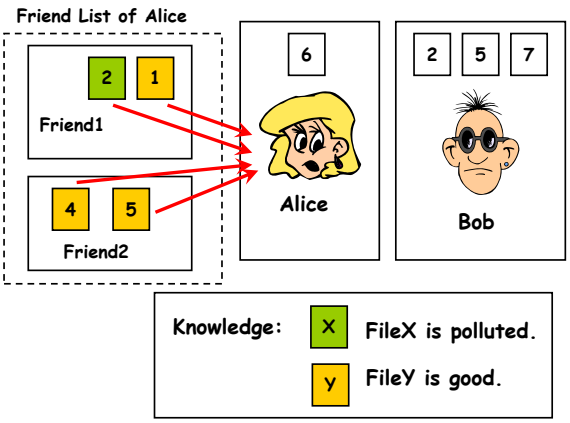

Fig. 1: Friend-based Scheme

reputation models can not provide a reasonable initial reputation value for the new member [5]. Due to lack of enough experiences, a new peer is easy to be deceived. However, in Soc, a new-comer can establish his friend relationship quickly with the friend-based scheme. Even if a new peer does not have any realistic friend in system, he will still be able to establish his friend relationships through joining friend-communities of current networks. Thus, the new peer can make use of the coincident knowledge of both his friends and the respondent to *detect* whether the latter is a deceiver. On the other hand, a new-comer can also detect directly the other peers. As asymmetrical information, for any peer who is challenged, he can not distinguish the challenge is sent from a new peer or an experienced peer. Thus, he dares not return a deceptive response.

3.2 Challenge-response Mechanism of Soc

Challenge-response mechanism is the important part of Soc. To elaborate the mechanism clearly, we will describe the whole process of our approach with the instance in Fig. 2.

Challenge Message: As shown in Fig. 2, *ChallengeMessage1* is comprised of some challenge messages. Each challenge message sent from *Alice* is an query for the evaluation of a file. And the one filled by *Bob* is the response for the evaluation of the file.

Challenge-response Mechanism: We present the details of challenge-response mechanism with the instance of Fig. 2. When *Alice* launches a search for *File3*, the system will return some relevant results. If the *File3* is owned by *Alice*'s friends, she can get the related results about the one from her friends. However, in the most cases, the owners of the file are not our friends. Here, Soc will perform challenge-response mechanism. In the instance of Fig. 2, due to *Bob* has *File3*, Soc will challenge *Bob*.

– In ***Step 1***, Soc chooses some coincident files between *Alice*'s friends and *Bob* to make challenge messages, and generates *ChallengeMessage1*. Soc inserts the chal-

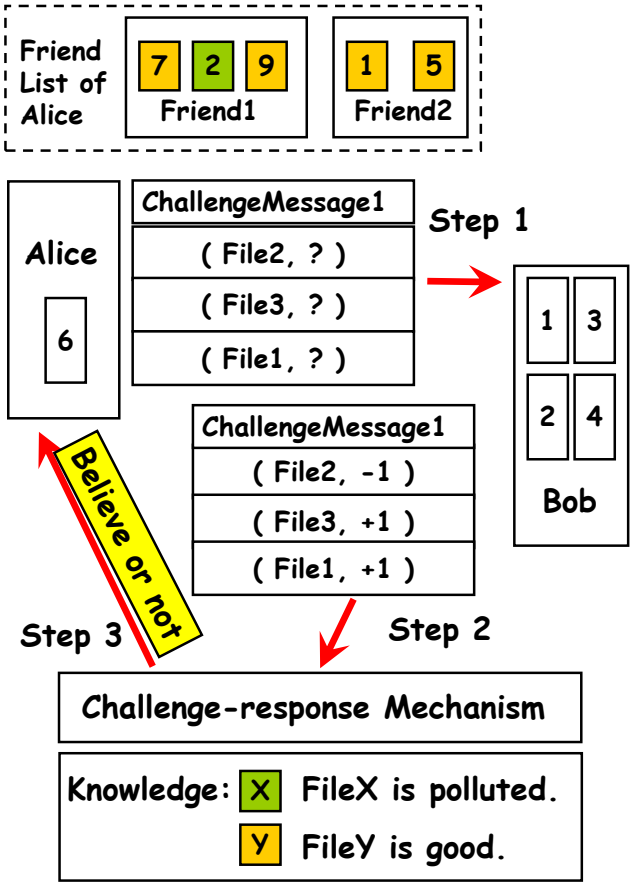

Fig. 2: Challenge-response Mechanism

lenge message made for the evaluation of *File3* into *ChallengeMessage1* randomly and then sends *ChallengeMessage1* to *Bob*.

– In ***Step 2***, *Bob* fills the responses about files in *ChallengeMessage1* and then returns.
– In ***Step 3***, challenge-response mechanism will tell *Alice* whether to believe *Bob*'s feedback according to percentage of correctness of his responses. The percentage can be set based on the different requirements of applications.

References [21], [22], and [23] indicate that the probability of the coincident files between peers in fire-sharing system is high, so in most cases Soc can find the coincident knowledge between challenger's friends and the respondent. For the peers who do not provide response, Soc will treat them as their feedbacks are mistake.

Penalty Mechanism for Deceivers: In Soc, each peer maintains a respondent list which stores the identifier of the respondent who has been challenged by the list owner

and a score for the respondent. After each response, challenger updates respondent's score according to the feedback of challenge-response mechanism as follows:

$$score = \begin{cases} \min(1, score + r) & \text{if return } \textbf{\textit{believe}} \\ \max(-1, score - p) & \text{otherwise} \end{cases} \quad (1)$$

Where, r and p represent the recompense and penalty given to respondent, respectively. We propose to set $p > r$, thus the score decreases faster then it increases. For a strange respondent, his initial score is 0. As shown in formula (1), if a deceptive peer is detected, his score will be decreased quickly.

When we launch a search, the responses from peers are ranked according to respondents' scores. Thus, the response from a deceptive peer will certainly be placed at the end of our search result. Through degrading rank, Soc reduces the probability of impact brought by deceptive peers.

Besides above penalty mechanism, when receiving a search request from the peer who has the negative score, we will ignore the request.

Effectiveness: Soc can address the problem of deceptive peers with active challenge-response mechanism. When a deceiver receives some challenges, he can not distinguish which message is the one that challenger really wants to know because of asymmetrical information. Thus, once returns the deceptive answer, the deceiver will be detected easily by Soc.

3.3 Practical Issues

In practice, Soc provides some robust solutions to handle the exceptive problems.

Deceptive Friend: We define the deceptive friend as the peer who provides the mendacious knowledge after being added into our friend list. In practice, the peer with many deceptive friends will be tampered with extended aggregations. In some sense, the peer will "pay the price" for having many deceptive friends. To address the problem of deceptive friend, Soc will perform the challenge-response mechanism to his friends intermittently. Thus, those deceptive friends will be find out under challenge-response mechanism.

Lack of Coincident Knowledge: If there is no any coincident knowledge between our friends and all respondents, what should be done? Soc uses *majority voting* to solve this exception. Here we take P2P file-sharing system as an example to describe the whole process of addressing problem. *First*, Soc will randomly select some files from each respondent and generate challenge messages combining with the file we are willing to know. *Second*, Soc sends those messages to corresponding respondents respectively. And *final*, we can judge the evaluation of file according to the majority voting result from all respondents. In fact, the capability of Soc will not be affected by this exceptive problem. Because of asymmetrical information, all the respondents can not know there is actually no coincident knowledge between challenger's friends and them. Thus, they dare not return deceptive responses.

PARAMETER	*VALUE*	
# File Titles *T*	200	
# File Versions *V*	500	
# Good Versions	30	
# Average Number of Friends	10	
# The Penalty Factor *p*	0.2	
# The Recompense Factor *r*	0.1	
# The Numbers of Friend Communities	100	
PARAMETER	*GOOD*	*DECEIVER*
# Peers	600	400
# Shared Files (startup)	30	90
# Deceptive Response	0-20%	70%-100%
Download / Day	0-5	0

Fig. 3: Experiment Parameters

3.4 Summary

In this section, we introduce Soc, a robust approach to address both the problems of deceptive peers and cold start in P2P networks. Especially for the practical issues, Soc proposes the reasonable solutions.

For different categories of P2P systems, Soc is a general approach. For instance, in a P2P file-sharing system, the knowledge is the evaluation of a file. When we want to download a file, we can challenge the file providers and decide whether to download this file according to the suggestion of Soc. Recently, the P2P tagging system [24] becomes very popular; however *spam tag* is a serious problem of the system. Soc also can be implemented in this application. For this instance, the knowledge we want to know becomes whether the tag is spam. We need to send some challenges to the users who post the tag; whether it is the spam tag will be detected easily.

4 Evaluation

This section presents our experiments and evaluations for Soc. As described in Section 3.4, Soc can be implemented in many different categories of P2P applications. Due to space constraints, here we only present the experiments of Soc in P2P file-sharing system. Section 4.1 will present our simulation environments. The main results will be discussed in Section 4.2.

4.1 Simulation Environment

Network Model: The simulation environment can be built by following main characteristics. Because the focus of our experiment is on the dissemination of files in P2P

networks, we do not need to consider any specific network architecture. All the messages routing is assumed perfect and transfer time can be negligible.

File Model: In our environment, there are T unique *titles*, each of which has V different *versions*[1]. The number of such versions may vary over time due to participants' downloading. At the simulation startup, files shared by peers are first selected by title, then its version. Both selections follow Zipf distributions with parameter $\alpha = 0.8$ [22]. Throughout our simulation, files to be downloaded are selected by first choosing the titles according the the same Zipf distribution, and then choosing the version according to the specific mechanisms respectively.

Peer Model: There are two categories of peers in simulation. They are good peers and deceptive peers. At startup, deceptive peers share polluted files only and good peers share only good ones. In experiments, the good peer will download file, leave and rejoin system and the percentage of their deceptive response is set below 20%. By contrast, the percentage of answers of the deceptive peers is set above 70%. Note that deceptive peers never leave the system nor download any files. Each peer has the different numbers of friends based on widely adopted Kleinberg model [26].

4.2 Experimental Results

We ran the experiment with many different configurations and get the similar results. The details for parameters of results discussed here have been shown in Fig. 3. Due to space constraints, we only present the capability of Soc for dealing with both the problems of deceptive peers and cold start.

Fig. 4 shows the rate of downloads for good file versions under the conditions that the percentage of peer feedback is 100%, 75% and 50% respectively. It is clear that, when the percentage of response of peers is 100%, the rate of good downloading can quickly converge above 90% in only 10 days. Even the percentage decreases to 50%, Soc can still turn the rate of good downloading to above 0.9 in 25 days. Moreover, we add the number of collusive deceivers in our experiments and the results are similar as Fig. 4 shown. The experiments prove that our approach can resolve the collusive deceivers effectively.

In order to illustrate the capability for dealing with the peer of cold start of Soc, we add 100 peers of cold start into current network. At startup, each of these 100 new peers has only one friend who has invited him into the system. Because of the mechanism of the friend-community, these peers can establish their own friend relationship very quickly. Fig. 5 shows that when the percentage of peer feedback is 100%, the average percentage of being deceived for these new peers can sharply drop to 10% in about 11 days. Thus it also proves that Soc can work well for the cold start problem.

5 Conclusion and Future Work

Current Peer-to-Peer networks are highly vulnerable to deceptive peers. Soc is a novel approach for combating the deceptive peers in P2P networks. Because of introducing

[1] The definition of the term title and version can be found in [25]

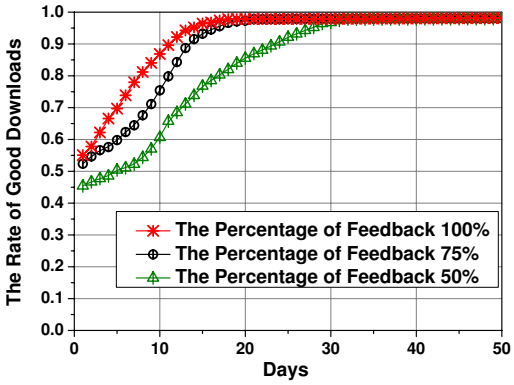

Fig. 4: Capability of Soc against Decoy Insertion in P2P File-sharing System

the friend-based scheme to P2P networks, it can construct the asymmetrical information between peers with the secret information of friend. In Soc, the peer can detect whether the other side is a deceiver through making use of the coincident knowledge of both his friends and the respondent. Moreover, Soc provides the penalty mechanism which can reduce the probability of impact brought by deceptive peers. Compared with existing reputation models, Soc addresses both the problems of collusion and cold start effectively. Soc can be implemented in many categories of P2P applications, for instance, P2P file-sharing system and P2P tagging system. We are currently studying how to introduce the game theory to model the strategy of peers and how to optimize Soc through analyzing the strategy of deceiver at the point of game theory.

References

1. Zhai, E., Chen, R., Wolinsky, D.I., Ford, B.: An untold story of redundant clouds: making your service deployment truly reliable. In: Proceedings of the 9th Workshop on Hot Topics in Dependable Systems, HotDep 2013, Farmington, Pennsylvania, USA, November 3, 2013. (2013) 3:1–3:6
2. Zhai, E., Chen, R., Wolinsky, D.I., Ford, B.: Heading off correlated failures through independence-as-a-service. In: 11th USENIX Symposium on Operating Systems Design and Implementation, OSDI '14, Broomfield, CO, USA, October 6-8, 2014. (2014) 317–334
3. Zhai, E., Sun, H., Qing, S., Chen, Z.: Sorcery: Overcoming deceptive votes in P2P content sharing systems. Peer-to-Peer Networking and Applications **4**(2) (2011) 178–191
4. Zhai, E., Li, Z., Li, Z., Wu, F., Chen, G.: Resisting tag spam by leveraging implicit user behaviors. PVLDB **10**(3) (2016) 241–252
5. Resnick, P., Kuwabara, K., Zeckhauser, R., Friedman, E.: Reputation systems. In: Communications of the ACM. (2000)
6. Xiong, L., Liu, L.: Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE Trans. Knowl. Data Eng. (2004)

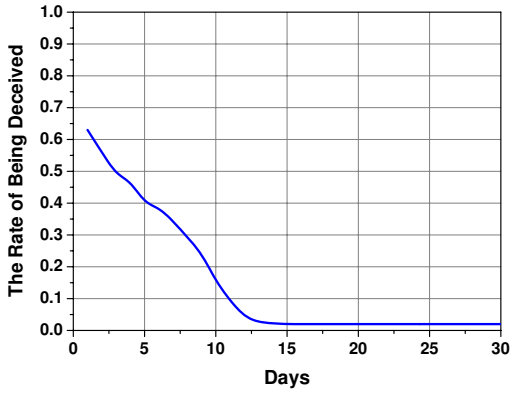

Fig. 5: Capability of Soc against the Cold Start Problem

7. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The eigentrust algorithm for reputation management in p2p networks. In: WWW. (2003)
8. Costa, C.P., Soares, V., Almeida, J.M., Almeida, V.: Fighting pollution dissemination in peer-to-peer networks. In: SAC. (2007)
9. Walsh, K., Sirer, E.G.: Experience with an object reputation system for peer-to-peer filesharing (awarded best paper). In: NSDI. (2006)
10. Zhai, E., Gu, L., Hai, Y.: A risk-evaluation assisted system for service selection. In: 2015 IEEE International Conference on Web Services, ICWS 2015, New York, NY, USA, June 27 - July 2, 2015. (2015) 671–678
11. Zhai, E., Chen, R., Cai, Z., Zhang, L., Lua, E.K., Sun, H., Qing, S., Tang, L., Chen, Z.: Sorcery: Could we make P2P content sharing systems robust to deceivers? In: Proceedings P2P 2009, Ninth International Conference on Peer-to-Peer Computing, 9-11 September 2009, Seattle, Washington, USA. (2009) 11–20
12. Zhai, E., Sun, H., Qing, S., Chen, Z.: Spamclean: Towards spam-free tagging systems. In: Proceedings of the 12th IEEE International Conference on Computational Science and Engineering, CSE 2009, Vancouver, BC, Canada, August 29-31, 2009. (2009) 429–435
13. Liu, B., Zhai, E., Sun, H., Chen, Y., Chen, Z.: Filtering spam in social tagging system with dynamic behavior analysis. In: 2009 International Conference on Advances in Social Network Analysis and Mining, ASONAM 2009, 20-22 July 2009, Athens, Greece. (2009) 95–100
14. Wang, Y., Zhai, E., Cao, C., Xie, Y., Wang, Z., Hu, J., Chen, Z.: Dspam: Defending against spam in tagging systems via users' reliability. In: 16th IEEE International Conference on Parallel and Distributed Systems, ICPADS 2010, Shanghai, China, December 8-10, 2010. (2010) 139–146
15. Wang, Y., Zhai, E., Hu, J., Chen, Z.: Claper: Recommend classical papers to beginners. In: Seventh International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2010, 10-12 August 2010, Yantai, Shandong, China. (2010) 2777–2781
16. Zhai, E., Ding, L., Qing, S.: Towards a reliable spam-proof tagging system. In: Fifth International Conference on Secure Software Integration and Reliability Improvement, SSIRI 2011, 27-29 June, 2011, Jeju Island, Korea. (2011) 174–181

17. Sun, C., Zhai, E., Chen, Z., Ma, J.: A multi-compositional enforcement on information flow security. In: Information and Communications Security - 13th International Conference, ICICS 2011, Beijing, China, November 23-26, 2011. Proceedings. (2011) 345–359
18. Zhai, E., Shen, Q., Wang, Y., Yang, T., Ding, L., Qing, S.: Secguard: Secure and practical integrity protection model for operating systems. In: Web Technologies and Applications - 13th Asia-Pacific Web Conference, APWeb 2011, Beijing, China, April 18-20, 2011. Proceedings. (2011) 370–375
19. Wang, Y., Zhai, E., Lua, E.K., Hu, J., Chen, Z.: isac: Intimacy based access control for social network sites. In: 9th International Conference on Ubiquitous Intelligence and Computing and 9th International Conference on Autonomic and Trusted Computing, UIC/ATC 2012, Fukuoka, Japan, September 4-7, 2012. (2012) 517–524
20. Jiang, J., Ding, L., Zhai, E., Yu, T.: Vrank: A context-aware approach to vulnerability scoring and ranking in SOA. In: Sixth International Conference on Software Security and Reliability, SERE 2012, Gaithersburg, Maryland, USA, 20-22 June 2012. (2012) 61–70
21. Gummadi, K., Dunn, R., Saroiu, S., Gribble, S., Levy, H., Zahorjan, J.: Measurement, modeling, and analysis of a peer-to-peer file-sharing workload. In: ACM SOSP. (2003)
22. Saroiu, S., Gummadi, P., Gribble, S.: A measurement study of peer-to-peer file sharing systems. In: MMCN. (2002)
23. Liang, J., Kumar, R., Xi, Y., Ross, K.W.: Pollution in p2p file sharing systems. In: INFOCOM. (2005)
24. Görlitz, O., Sizov, S., Staab, S.: Pints: Peer-to-peer infastrubture for tagging systems. In: IPTPS. (2008)
25. Liang, J., Naoumov, N., Ross, K.W.: Efficient blacklisting and pollution-level estimation in p2p file-sharing systems. In: AINTEC. (2005)
26. Kleinberg, J.M.: The small-world phenomenon: an algorithm perspective. In: STOC. (2000)