Sorcery: Could We Make P2P Content Sharing Systems Robust to Deceivers?

Ennan Zhai[†,*], Ruichuan Chen[‡,*], Zhuhua Cai[‡,*], Long Zhang[‡,*]
Eng Keong Lua[§], Huiping Sun[†,*], Sihan Qing[†], Liyong Tang[‡,*], Zhong Chen[†,‡,*]
[†]School of Software and Microelectronics, Peking University, China
Email: zhaien@infosec.pku.edu.cn, {sunhp, qsihan, chen}@ss.pku.edu.cn
[‡]Institute of Software, School of EECS, Peking University, China
Email: {chenrc, caizh, zhanglong, tly}@infosec.pku.edu.cn
[§]Carnegie Mellon University, USA
Email: englua@cmu.edu
[*]Key Laboratory of High Confidence Software Technologies, Ministry of Education, China

Abstract

*Deceptive behaviors of peers in Peer-to-Peer (P2P) content sharing systems have become a serious problem due to the features of P2P overlay networks such as anonymity, self-organization, etc. This paper presents **Sorcery**, a novel active challenge-response mechanism based on the notion that one side of interaction with dominant information can detect whether the other side is telling a lie. To make each client obtain the dominant information, our approach introduces social network to the P2P content sharing system; thus, the client can establish friend-relationships with peers who are either acquaintances in reality or those reliable online friends. Using the confidential voting histories of friends as own dominant information, the client can challenge the content providers with the overlapping votes of both his friends and the content provider, thus detecting whether the content provider is a deceiver. Moreover, Sorcery provides the punishment mechanism which can reduce the impact brought by deceptive behaviors, and our work also discusses some key practical issues. The experimental results illustrate that Sorcery can effectively address the problem of deceptive behaviors, and work better than the existing reputation models.*

1 Introduction

Peer-to-Peer (P2P) content sharing systems, e.g., BitTorrent [1], KaZaA [3], have become increasingly popular. However, due to the anonymity and self-organization nature of P2P overlay networks, the participants have to face some potential risks involved in the application transactions without adequate experience and knowledge about other users. Many studies indicated that P2P content sharing systems are highly vulnerable to *deceptive behaviors* [23, 28].

In a typical deceptive behavior, individual or collusive *deceivers* (the peers who deceive other participants) first publish lots of content that may contain invalid meta-data or corrupt data; meanwhile they also download some content from system. Then, they vote on the content in system incorrectly in order to mislead other users. Unable to distinguish authentic content from the corrupt one, the normal P2P users download the undesirable content into their folders. Thus, we say these normal users are deceived. As a result, the corrupt content spreads through the P2P network with an extraordinary speed, and causes the availability of content sharing systems to be low.

Generally, previous studies on addressing deceptive behaviors mainly focused on the reputation models. However, due to the nature of reputation models, such as passive aggregation of experiences, the client is easy to become a victim when encountering the collusive deceivers or individual tricky deceiver [25]. The above situation can be explained that the adversaries in the system sit on the *dominant position*, and the solution is that we need to achieve the conversion of the dominant position through constructing our own *dominant information*. The fundamental insight driving our work is that social network can help the users construct the confidential and reliable friend-relationships [22], and we may treat the confidential information (e.g., content, vote history, etc.) of friends as the dominant information, since the friend information is owned by the client only. Therefore, the client can detect whether the content provider is a deceiver, using the overlapping voting histories of the content of both the friends and the content provider.

Based on the above analysis, this paper introduces Sorcery, a novel *challenge-response* mechanism using social

network to construct the dominant information. The *challenge* denotes the query about the votes of some content, and the *response* denotes the response messages to "answer" the challenge (The details are mentioned in Section 3.2). Sorcery encompasses *three* key techniques to detect and punish the deceivers in P2P content sharing system:

Social Network: Sorcery introduces social network to the P2P content sharing system, and thus each client can establish his own friend-relationships. These friends share their own information (e.g., content, votes, etc.) with the client, and the friend information of the client is confidential to other peers in the system.

Challenge-response Mechanism: Sorcery clients utilize the overlapping voting histories of both his friends and the content provider to challenge the latter actively, and judge whether the other side is a deceiver or not based on the correctness of his response.

Punishment Mechanism: Sorcery clients rank each search result based on the honesty of the content providers; therefore, the probability of impact brought by deceivers is reduced.

We conduct simulation studies with different network, peer, content and execution models. The evaluation results show that Sorcery can effectively detect the deceivers, and make the users avoid from downloading the corrupt or malicious content. Throughout the entire experiments, we assess the performance of Sorcery as compared to the ingenious reputation model — Credence [27]. In addition, we also discuss the performance of Sorcery on fighting against the man-in-the-middle (MITM), Sybil [12], and Denial-of-Service (DoS) attacks in the end of this paper.

The rest of this paper is organized as follows. The related work will be given in Section 2. The details of Sorcery are described in Section 3. Section 4 presents the simulation methodology and evaluation results. Finally, we conclude with a discussion of incorporating Sorcery with the schemes against MITM, Sybil and DoS attacks in Section 5.

2 Related Work

Currently, many reputation models have been proposed to resist the deceptive behaviors in the P2P content sharing systems. In general, these reputation models can be grouped into three categories: *peer-based* reputation models, *object-based* reputation models and *hybrid* reputation models.

Peer-based Reputation Models: The past work of peer-based reputation models, e.g., EigenTrust [15], PeerTrust [28] and Scrubber [9], relies mainly on assigning the reputation score based on performance measures of peers, and then judge the deceptive peers according to the peers' reputation scores.

Object-based Reputation Models: Credence [27] is the ingenious object-based reputation model. It enables the clients to weigh votes according to the statistical correlation between the client and his peers, thus reducing the probability of believing the votes of deceivers.

Hybrid Reputation Models: Aiming at combining the benefits of both peer-based and object-based reputation models, several hybrid reputation models, e.g., XRep [11], X^2Rep [10] and Extended Scrubber [8], have been further presented. XRep and X^2Rep extend the work in [7] by additionally computing the reputation of object with the weights based on the past voting behavior of peers.

Besides the reputation models, the micropayment techniques can also be used to confront deceptive behaviors by imposing a large cost on the deceivers, e.g., MojoNation [4]. Furthermore, the fair exchange protocol [14] provides the mechanism similar with the micropayment to eliminate the benefit gained by deceivers in P2P systems.

To the best of our knowledge, none of the previous work focused on using the challenge-response-like approach to address the problem of deceptive behaviors in the P2P content sharing systems.

3 Sorcery

In a typical P2P content sharing system, clients publish some *content*, and each content has a specific *title*. Normally, a title has various *versions*[1], each of which is shared by a group of *providers*. Without loss of generality, this paper defines content item as the specific version associated with a designated title. Moreover, the peers who have voted on a specific content item are called the content item's *content voters*. Note that the content provider may have not voted on some of his shared content items; also, the content voter may have removed the content items which were voted by himself.

Generally, a client issues queries to the system in the form of keywords. The providers respond by sending the matched content items back to the client. Here, the client needs to judge the authenticity of each content item before attempting to obtain it, since some may contain malicious or corrupt data. Due to the lack of reliable evidence for reference, a client usually makes the downloading decision based on the votes on this content item.

Sorcery helps the clients judge the authenticity of votes on the *target content items*, and reduces the malicious impact brought by the deceivers. Before giving the design rationale, we first describe the infrastructure of Sorcery.

Infrastructure of Sorcery: Sorcery introduces the social network to the P2P content sharing system, and thus

[1]The definition of the terms title and version can be found in [19]

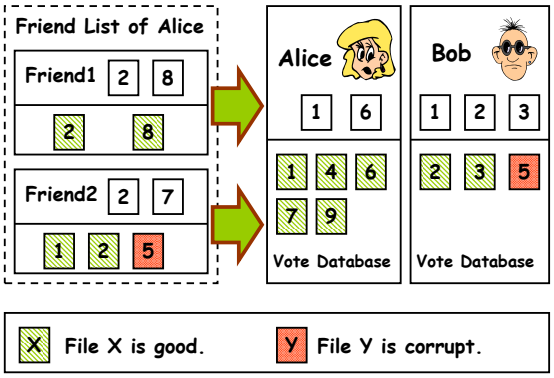


Figure 1: Friend-relationships. To simplify the description, the $File$ represents the specific content item, and we denote the content item X as $FileX$.

each client may establish the friend-relationships with some peers in the system. A Sorcery client needs to maintain two lists: one is *friend list* which contains the information of the client's friends (as shown in Figure 1), and another one is *respondent list* which is comprised of the peers who have ever been challenged by the client. Note that the client's friend list can not be seen by other peers. Using the confidential friend information, the client constructs the dominant information, which is reflected during the challenge-response (Section 3.2), with respect to the other peers in the system. Moreover, each peer maintains a *vote database* to store the peer's own votes.

3.1 Social Network

Sorcery introduces the social network to the P2P content sharing systems. In this section, we describe the details of the social scheme of Sorcery.

Friend List: In Sorcery, each client has many friends and stores their information in his own friend list. The friend list of client is confidential to other peers in the system; therefore, Sorcery constructs the dominant information for each client with the confidential friend information. As shown in Figure 1, $Alice$ publishes $File1$ and $File6$, and moreover her votes on $File1$, $File4$, $File6$, $File7$ and $File9$ are stored in her local vote database. In this instance, $Alice$ has two friends in her friend list, and the friends' information, e.g., content, votes, etc., can be seen by $Alice$, but Bob cannot obtain any information of the $Alice$'s friends. Moreover, $Alice$ will timely update the information of own friend list.

Establishment of Friend-Relationships: Any peer can be invited by an existing participant in the network, and thus added into the system; meanwhile the inviter will become the friend of the newcomer automatically. The invitation scheme ensures that the client at least owns one friend in the system. Besides the method of invitation, the client can establish friend-relationships with the peers who are the real-world acquaintances, or the online friends recognized in other social networks. For Sorcery, the friend-relationship is symmetric, and 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. The fundamental reason of utilizing the friends' information is that they are much more trustworthy than the anonymous peers in the system; however, the client's friends may be malicious or compromised, thus we will present a mechanism to address this practical issue, in Section 3.4.2.

Effectiveness: Due to introducing social network to the P2P content sharing system, Sorcery can address the problem of *cold start*. This problem means that when a new peer joins in the system, he can easily be deceived due to the lack of enough experiences of interactions with others. The studies in [23, 25] indicated that the existing reputation models cannot completely address the cold start problem. However, in Sorcery, a newcomer can establish his friend-relationship quickly with the social network, and collect the voting histories from his friends. Therefore, the newcomer can quickly obtain the experiences as a mature participant in the system.

3.2 Challenge-response Mechanism

In this section, we describe the details of challenge-response mechanism. Besides the friend list, each client also maintains a respondent list which stores the *reliability degrees* of the response peers who have been challenged by the client. Based on honesty of the response peer, the client computes the reliability degree with respect to the response peer (The calculation of reliability degree will be given in Section 3.3).

After the client issues a search with some keywords, the system returns with the matched content items, and ranks the search results in descending order based on the reliability degrees of the providers. If the target content item is owned by the client's friends, he can download the content item from his friends directly. However, in the most cases, the client's friends do not have the target content item. Therefore, the client should choose some of target content providers, based on their orders, to perform challenge-response. To elaborate the process of challenge-response clearly, we define $\{Vote_{j(i)}\}_{i=1}^{n}$ as the set of overlapping votes of the client's friends and the provider j, where n denotes the size of the set of overlapping votes, and $Vote_{j(i)}$ denotes the vote of the provider j on the content item i. Then, we define $\{C_i\}_{i=1}^{n}$ as the set of content items associated with the elements of $\{Vote_{j(i)}\}_{i=1}^{n}$. The element of $\{C_i\}_{i=1}^{n}$, C_i, denotes the content item associated with the

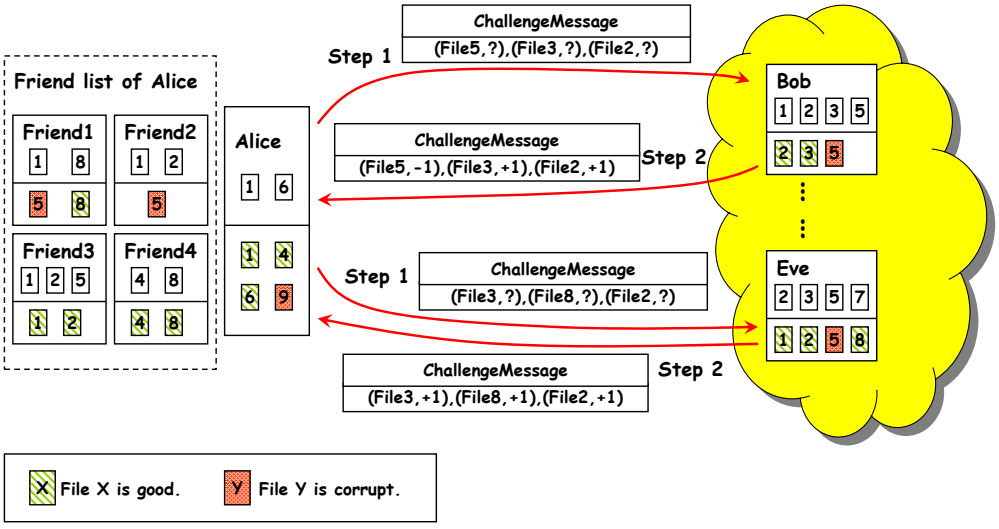


Figure 2: Challenge-response Mechanism. To simplify the description of instance, the $File$ represents the specific content item, and we denote the content item X as $FileX$.

$Vote_{j(i)}$. The process of challenge-response is as follows:

- **Step 1 — Challenge:** The client first generates the *challenge message* which is comprised of the queries for some content items. Here, the query denotes the voting request on a specific content item C_i. Meanwhile, the client inserts the query for the target content item into the challenge message randomly for confusing the target content provider. Note that the reason that we do not use the queries for all the elements of $\{C_i\}_{i=1}^{n}$ is to avoid the target provider from knowing the details of $\{Vote_{j(i)}\}_{i=1}^{n}$ and judging the client's target content item next time. Afterwards, the challenge message is sent to the target content provider. In a similar way, the client also generates the challenge message to challenge other providers of the target content item.

- **Step 2 — Response:** After receiving the challenge message, the provider should respond the client with the *response message* which is comprised of local votes for the queries in the challenge message. These votes contain the provider j's votes on each C_i and the target content item. Similarly, other providers also need to respond the challenges in the same way.

When the client receives the responses, he may estimate whether to believe each response peer's vote on the target content item based on ϑ — the rate of each response peer's correct answers. This rate can be set according to the different requirements of applications. Normally, we think $\vartheta \leq 0.5$ indicates weak or no correctness. Due to the feature of utilizing the correctness of response, in practical applications, Sorcery may be threaten by the target deceivers who falsely vote on the popular content items and vote correctly on other normal content items. In fact, the above threat will not harm Sorcery. Because Sorcery client first examines whether his friends own or have voted on the popular content item (target content item), we believe that, normally, the client's friends may have downloaded or voted on the popular content item. Therefore, the client could obtain directly the popular content item or the associated votes from his friends before challenging the providers of target content item.

Example: To elaborate the challenge-response mechanism more clearly, we describe the whole process with the instance in Figure 2. In this instance, $Alice$ issues a search for $File3$. Because $Alice$'s friends do not have $File3$, $Alice$ performs the challenge-response to some providers of $File3$ according to their orders. As shown in Figure 2, we assume that Bob and Eve are the top2 providers of $File3$ in $Alice$'s ranking result. Therefore, $Alice$ performs the challenge-response to them as follows:

- **Step 1 — Challenge:** $Alice$ chooses the queries for $File2$ and $File5$ to generate the challenge message, since $Alice$'s friends have the votes on the two content items. Then, Sorcery client inserts the query for $File3$ into the challenge message randomly, and sends to Bob; likewise, $Alice$ also generates the challenge message to challenge Eve.

- **Step 2 — Response:** After receiving the challenge message, Bob should "answer" the $Alice$'s challenge

with his own votes on $File2$, $File3$ and $File5$, and then returns the response message to $Alice$ as Figure 2 shown. Similarly, Eve also needs to respond the challenge of $Alice$ in the same way.

After $Alice$ receives the responses, the challenge-response mechanism will tell $Alice$ whether to believe the votes of Bob and Eve on $File3$. However, in some practical cases, there are possibly two issues as follows:

- For the peers challenged by the client, if they do not provide responses in a certain time internal, the client chooses to treat them as the deceivers.
- There are possibly no overlapping voting histories of both the client's friends and the content providers. We will discuss an approach to address this practical issue in Section 3.4.1.

3.3 Punishment Mechanism for Deceivers

In order to reduce the possibility of impact brought by deceivers, Sorcery proposes a severe punishment mechanism to the client for computing the reliability degrees of response peers. Using the punishment mechanism, the client i computes the reliability degree, $RD_{i(j)}$, with respect to the response peer j according to each judgemental result for the peer j as follows:

$$RD_{i(j)} = \begin{cases} \max(-1, RD_{i(j)} - pn^2) & \text{if } j \text{ is a deceiver} \\ \min(1, RD_{i(j)} + r) & \text{otherwise} \end{cases} \quad (1)$$

where

- n: The number of peer j's response being judged as deceptive behavior.
- p: The penalty factor given to the peer j.
- r: The recompense factor given to the peer j.

In Equation (1), if a deceptive peer is detected, his reliability degree will be decreased quickly. Sorcery allows the peers recover their reliability degrees by responding genuinely. We propose to set $p > r$, and thus the reliability degree can decrease faster than it increases. For a strange response peer, his initial reliability degree is set to 0.

Because each search result is ranked according to the content providers' reliability degrees, the content items provided by the deceivers will be placed at the end of search result. Therefore, by degrading the rank of the response content items, Sorcery reduces the impact brought by the deceivers. Besides the demotion-based punishment, when receiving a search request from the peer who has the negative reliability degree, the client should ignore the request.

To spread the impact of punishment, after computing the reliability degree, the client will broadcast the new reliability degree to his friends and friends-of-friends; meanwhile, the reliability degrees computed by both the client's friends and friends-of-friends can also be received by the client. Therefore, the power of punishment should be expanded in the reliable "social group". The study in [13] demonstrated both the reliability and security of the friends-of-friends in the real-world social networks. Therefore, we can believe the reliability degrees provided by the patulous friends.

3.4 Practical Issues

This section mainly discusses how to address three important practical issues:

- The client's friends do not have the overlapping voting histories with the content providers.
- The unreliable or compromised friends.
- The malicious user who correctly responds the challenge, but to transfer with bogus content item.

3.4.1 Lack of the Overlapping Votes

Due to the lack of common interests with the client's friends, the content providers possibly do not have the overlapping votes with the client's friends. The studies in [18, 26] indicated that it is a high proportion that most peers in the system have overlapping votes with the voters of any content item. Therefore, when the client's friends do not have the overlapping votes with the content providers, Sorcery client seeks for the target content voters, and ranks these voters. The ranking score of the client i with respect to the voter j, $RS_{i(j)}$, is computed as follows:

$$RS_{i(j)} = VN_j \times RD_{i(j)} \quad (2)$$

where

- VN_j : The total number of the votes of voter j.
- $RD_{i(j)}$: The reliability degree of the client i with respect to the voter j.

According to the order of voters, the client will challenge some of these voters. As shown in Figure 3, we assume that $Carol$ and $Dave$ are the top2 voters of $File3$ in the $Alice$'s ranking result, and they do not have the $File3$. According to the result of challenging $Carol$ and $Dave$, $Alice$ can judge that $Carol$ is a genuine user and $Dave$ is a deceiver. Thus, $Alice$ may believe the $Carol$'s vote on $File3$,

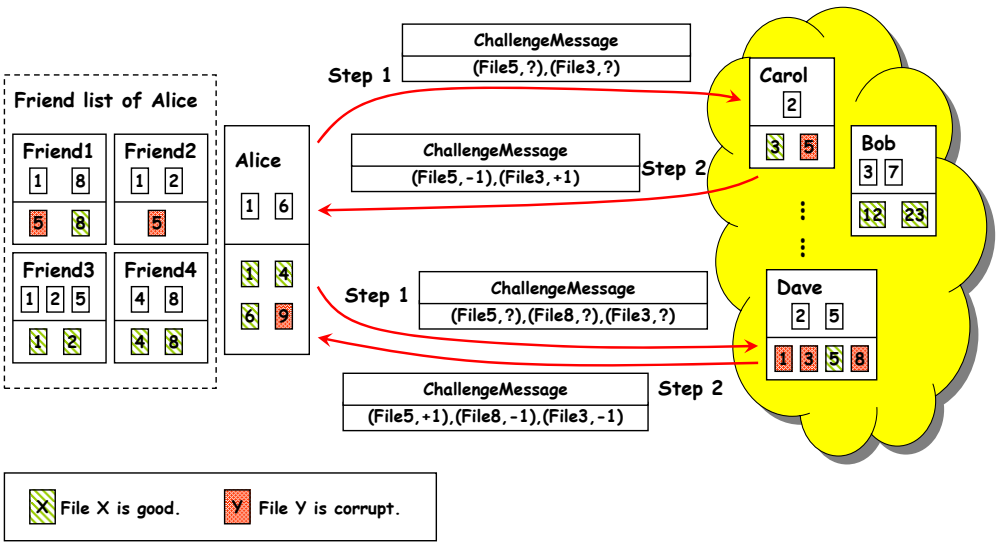


Figure 3: Challenge-response to the content voters. To simplify the description of instance, the $File$ represents the specific content item, and we denote the content item X as $FileX$.

and download $File3$ from the system; otherwise, $Alice$ will challenge other voters based on the order of voters.

Discussion: The studies in [26, 27] reported the interesting fact that, even for the voters of normal content items, any peer in system still has the overlapping votes with these voters with high probability. Therefore, we believe that the client's friends have the overlapping votes with the target voters, with relatively high probability. Even if there indeed exists the instance that the client's friends have no overlapping votes with all the target content voters, we propose that Sorcery client leverages the transitivity of social network, e.g., the friends-of-friends, to amplify the voting set used to generate the challenge. Moreover, Sorcery can also utilize the patulous reliability degrees (mentioned in Section 3.3) to help the client judge the authenticity of target content item. Note that the study in [13] has demonstrated the reliability for the friends-of-friends. To bound the harms incurred by the Sybil attacks during expanding social network, we adopt SybilLimit [29] based on the common insight on social network with our work. Our experiments also demonstrate that the expansion of social network indeed makes Sorcery more robust to the deceivers.

3.4.2 Unreliable Friends

In the practical applications, some friends may be online deceivers or compromised. Therefore, Sorcery proposes the similarity between the client i and his friend f, $Sim_{i(f)}$, to resist the harms incurred by unreliable friends. The calculation of the similarity is based on the cosine technique as follows:

$$Sim_{i(f)} = \frac{\sum_{k \in C}(V_{i(k)}V_{f(k)})}{\sqrt{\sum_{k \in C}(V_{i(k)})^2}\sqrt{\sum_{j \in C}(V_{f(j)})^2}} \quad (3)$$

where

- $V_{x(y)}$: The vote from peer x to the content item y, and normally the value is $+1$ or -1. If x has not voted on the content item y, the value is 0.
- C : The content set of the system.

For two peers who have not voted on the same content items, the similarity between them is set to 0. The client will compute the similarities of all his friends intermittently. Once this friend has the similarity lower than 0.5, the client should not utilize the votes of the friend to generate the challenge message, since the study in [26] indicated that the similarity lower than 0.5 represents weak or no correlation between two peers.

3.4.3 Incredible Interaction

Another serious security vulnerability is that, in the process of challenge-response, a malicious content provider may correctly respond the client's challenge, but replies with the bogus content item. This threat has been mentioned in [28] called *incredible interaction*, and Sorcery proposes the similar approach with [28], but based on the social network, to bound the impact of incredible interaction.

When the client i wants to download the content item from the provider j, he first examines his own evaluation history of interaction. If i has interacted with j, he may make decision based on his past experiences; otherwise, the client i should compute the *credibility* with respect to j, $Cred_{i(j)}$, as follows:

$$Cred_{i(j)} = \frac{\sum_{f \in F}(Sim_{i(f)} Eval_{f(j)})}{|F|} \quad (4)$$

where

- F : The set comprised of i's friends and his friends-of-friends who have evaluated the interaction with the peer j.
- $Eval_{f(j)}$: The evaluation of the interaction from f to j, and the value is $+1$ or -1, where $+1$ denotes a satisfied interaction and -1 denotes the bogus one.
- $Sim_{i(f)}$: The similarity between the client i and f.

The credibility is interpreted as an estimate of the authenticity of the content provider. This estimate should be used to make a decision to accept or reject downloading. Due to using the similarity as the weight to compute the credibility, the evaluations of the malicious friends cannot make harm. After downloading, the client needs to evaluate this interaction. These evaluations are shared to his friends and the friends-of-friends; meanwhile, the client may also obtain the evaluations shared by his friends and friends-of-friends.

Discussion: The threat of incredible interaction is difficult to avoid by most existing schemes. To the best of our knowledge, most reputation models, such as EigenTrust [15] and Credence [27], cannot address this problem. On the other hand, Sorcery utilizes reliable social network to bound this threat, and adding integrity verification into the overlay network will be our future work.

4 Evaluation

In this section, we first describe the simulation setup, and then present the performance metric. Finally, we evaluate Sorcery as compared with the Credence [27].

4.1 Simulation Setup

To evaluate the performance, we developed a P2P content sharing prototype system with all the mechanisms of Sorcery. Moreover, we generate several models with different parameters — follow the certain distributions. Table 1 shows the important parameters throughout our simulations.

Network Model: In the following simulations, we choose Gnutella [2] as the underlying overlay network of Sorcery. Because our focus is on the dissemination of the content in the network, we assume the perfect overlay routing and content discovery. Furthermore, the transfer time is assumed to be negligible.

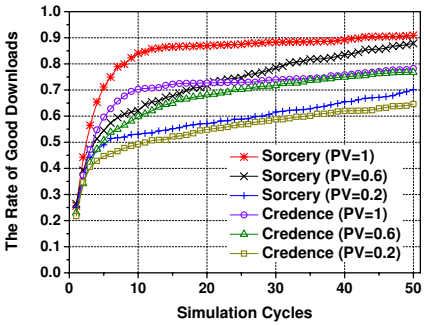
Table 1: Experimental Configurations

Parameter	Meaning
PC	The correctness rate that the genuine peer votes on the content items.
PD	The probability that the deceiver votes on the content items correctly.
PR	The probability that the peer gives response when he is challenged.
PV	The probability that the peer gives votes on the content items.
FC	The total proportion of collusive deceivers.
FD	The total proportion of the deceivers.
FN	The number of each client's friends.

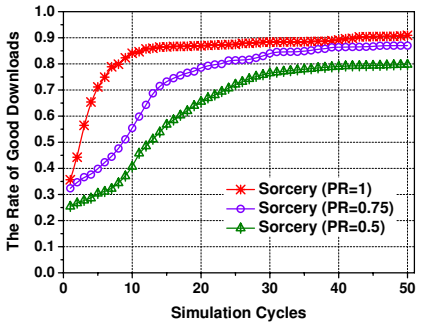
Peer Model: The network is composed of $5,000$ peers, and there are two categories of peers, genuine peers and deceivers, in our simulation. At the startup, the genuine peers only publish the good content items, and correctly vote on the content items; whereas, the deceivers share the corrupt content items, and give the positive votes on them. Throughout our simulation, two categories of peers may download content items, leave and rejoin the system. We generate the social network of simulation according to small world property of online social networks [20], and establish the friend-relationships for the peers based on the widely adopted Kleinberg model [16].

Content Model: The study in [18] indicated the existence of a large number of corrupt versions for the single file (title) in the actual system. In our simulations, there are $1,000$ unique files (titles), and each of which has 500 different versions containing 50 good versions. At startup, each genuine peer publishes 30 content items and each deceiver shares 200 content items. Furthermore, the versions published by a peer are determined by first selecting a certain title and then its version. Specifically, both selections follow Zipf distribution with the parameter $\alpha = 0.8$ [18].

Execution Model: Different queries are initiated at uniformly distributed peers in the overlay network. An experimental simulation is composed of 50 simulation cycles. In each cycle, the selection of $0-5$ specific content items to download is done by first selecting a title and then choosing a version based on the mechanisms of Sorcery. After each simulation cycle, the number of corrupted downloads is calculated. The genuine peers and deceivers may download good and corrupt content items; however, the deceivers should give the positive vote on a corrupt content item and a negative vote on the good content item. On the other hand,



(a) Impact of the Probability of Peers Voting.



(b) Impact of the Probability of Peers Responding.

Figure 4: Impact of Cooperation among Peers

a genuine peer should give the votes, with the correctness PC, on the content items in the system. Without the especially emphasized, we set $PC = 0.9$, $PD = 0$, $PR = 1$, $PV = 1$, $FC = 0$, $FD = 0.2$, and $FN = 6$ as the default configurations of our simulations. Each experimental simulation is run 5 times and the results of all runs are averaged.

4.2 Performance Metric

In the following simulations, we characterize the system performance with the *rate of good downloads*. It is defined as the rate of downloads that the clients acquire good content items in one simulation cycle. Specifically, this metric is computed at the end of each simulation cycle.

4.3 Experimental Results

Recently, most of the defense mechanisms against deceptive behaviors deployed in P2P content sharing systems are based on the reputation models. Among these models, Credence [27] is an ingenious model deployed on a real-world network and, moreover, its scenario is similar to Sorcery. Therefore, we compare the performance of Sorcery with that of Credence throughout our experiments.

Impact of Cooperation among Peers: Due to the feature that both Sorcery and Credence need the votes from the peers, we compare Sorcery with Credence under the different PVs. As shown in Figure 4a, when the PV is set to 1.0, the rate of good downloads of both Sorcery and Credence can increase to higher than 0.7 after 5 and 10 cycles respectively. However, when setting PV to 0.2, we notice that both the performances of Sorcery and Credence are affected by this low probability of peer voting — the rates of good downloads of two models are always lower than 0.7 during 50 simulation cycles. Therefore, we conclude that these two models strongly depend on the cooperation of peer voting, and Sorcery can work with the better convergence due to the reliable votes provided by the clients' friends.

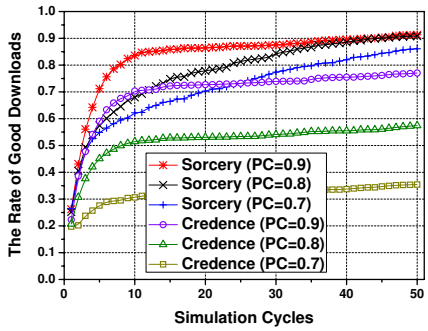
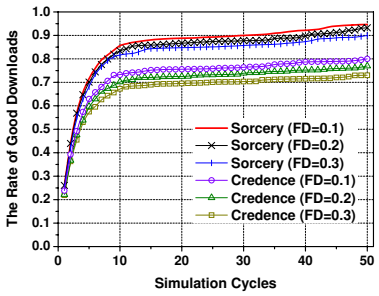


Figure 5: Impact of the Correctness of Voting

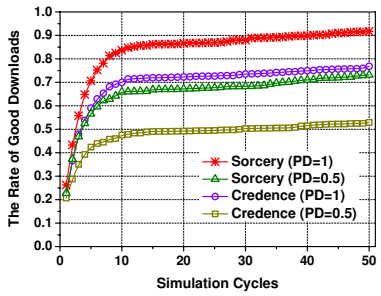
Figure 4b shows the rate of good downloads of Sorcery under the conditions that PR is set to 1.0, 0.75 and 0.5 respectively. It is clear that, when $PR = 1.0$, the rate of good downloads can quickly converge to 0.8 in only 8 cycles. Even if PR is decreased to only 0.5, Sorcery can still turn the rate of good downloads to above 0.7 after 20 cycles. This experimental result indicates that, in order to construct the good download rate of content in the system, the peers should actively respond each other.

Impact of the Correctness of Voting: In this simulation, we evaluate the performance of Sorcery compared with Credence under different correctness rates of the genuine peers' votes (PC). As the results shown in Figure 5, it is clear that, the change of PC makes a strong influence on both the performances of Sorcery and Credence. We notice that Sorcery can always work better than Credence under three different values of PCs. Therefore, we conclude that Credence is more vulnerable to PC than Sorcery.

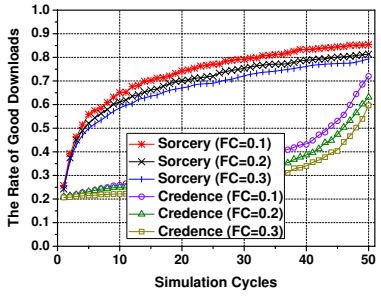
Impact of the Deceivers: The simulation in Figure 6a



(a) Impact of the Normal Deceivers.



(b) Impact of the Tricky Deceivers.



(c) Impact of the Collusive Deceivers.

Figure 6: Impact of the Deceivers in System

simulates the impact of total proportion of deceivers (FD) on the performances of the simulated two models. Throughout the simulations, we assume the system will not be attacked by P2P worms, so that, the percentage of deceivers should generally not be higher than 30%. Interestingly, the result in Figure 6a shows that the different FDs do not significantly influence the performances of Sorcery and Credence. The reason is that, although the proportion of deceivers is increased, Sorcery client still utilizes the votes of his own friends to challenge other peers respectively. Therefore, the performance of Sorcery cannot be affected by the proportion of deceivers. For Credence, because peers' downloads are based on their own judgements of the content authenticity, Credence clients cannot be affected along with the increase of deceivers.

In Figure 6b, we evaluate the performances of Sorcery and Credence against the tricky deceivers who can pretend the genuine peers by voting correctly on some content items, and deceive the normal users. Figure 6b indicates that two models both have been affected when setting PD to 0.5. The results shown in Figure 6c demonstrate, under the collusive attacks, Sorcery can work much better than Credence. From these two experiments, we deduce that the key reason that makes Sorcery outperform Credence is to introduce the social network into the P2P content sharing system. This result clearly demonstrates the conclusion in the study [22] — the social network can make the P2P sharing systems more robust.

Impact of Peer's Friend Number: The above simulations have reflected the impact of social network. In this simulation, we discuss the impact of each peer's friend number (FN). Figure 7 shows that, as setting the FN to 2, 4, 6 and 8 respectively, the performance of Sorcery changes a lot. When each peer only has two friends, the rate of good downloads is always lower than 0.4 during 50 simulation cycles; however, as setting FN to 6, Sorcery can perform robustly to the good downloads. Interestingly, when we vary the FN to 8, the enhancement of performance is not so prominent as the previous experiments (FN is set to 2, 4 and 6). This result indicates that a Sorcery client actually does not need to hold a large number of friends.

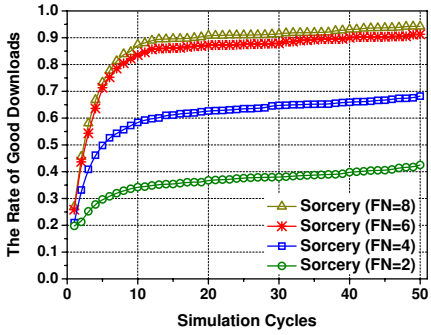


Figure 7: Impact of Peer's Friend Number

5 Conclusion and Discussion

Sorcery is a novel challenge-response approach against deceptive behaviors based on the confidential information extracted from social network. Our experimental results illustrate that Sorcery can effectively address the problem of deceptive behaviors, and work better than the existing reputation models. However, some other types of attacks can also be mounted against Sorcery, such as MITM attack, Sybil attack [12] and DoS attack.

Generally, an MITM attacker could read, insert and modify the messages between two sides of the challenge-response without letting any of them has the knowledge of compromised transaction between them. Therefore, both the challenge and response generated by the two sides may be unauthenticated. To resist such MITM attack, the peers should dynamically maintain a trusted group to perform multiple independent exchanges originating from the different trusted group members (similar to the mechanism de-

scribed in [6]); then, the peer can execute the Byzantine agreement protocol [17] to obtain the actual messages.

Another important security vulnerability is the Sybil attack — a deceiver takes on multiple identities and pretends to be distinct peers [12]. Under Sybil attacks, the challenge-response mechanism is easy to be compromised. Based on the same insight, Sorcery clients could directly utilize their social networks to limit the deceptive behaviors of Sybil attackers as the approaches in studies [29, 30]. As an alternative, we could also adopt the computational puzzle scheme against the Sybil attack [5, 24].

In general, the DoS attack is the serious threat where one or more users attempt to thwart genuine users from having access to legitimate services. In our work, the popular content providers may be flooded by huge amounts of challenge, i.e., Sorcery may suffer from the DoS attack. The study [21] indicated that the ingenious solution against DoS attack is based on the calculation of puzzle scheme. Therefore, to resist DoS attack, each Sorcery client may compute moderate expense, but not intractable puzzles to gain the admission to challenge those popular content providers.

Could Sorcery make P2P content sharing systems robust to deceivers? So far, we see a robust Sorcery against the deceivers in the P2P content sharing systems.

Acknowledgment

We would like to thank the anonymous reviewers for their helpful suggestions. This work was supported in part by the NSFC under grant No. 60773163 and No. 60873238, as well as the NSFC & JSPS under grant No. 60911140102.

References

[1] BitTorrent. http://www.bittorrent.com.
[2] Gnutella. http://www.gnutelliums.com.
[3] KaZaA. http://www.kazaa.com.
[4] MojoNation. http://www.mojonation.net.
[5] N. Borisov. Computational Puzzles as Sybil Defenses. In *Peer-to-Peer Computing*, pages 171–176, 2006.
[6] R. Chen, W. Guo, L. Tang, J. Hu, and Z. Chen. Scalable Byzantine Fault Tolerant Public Key Authentication for Peer-to-Peer Networks. In *Euro-Par*, pages 601–610, 2008.
[7] F. Cornelli, E. Damiani, S. D. C. di Vimercati S. Paraboschi, and P. Samarati. Choosing Reputable Servents in a P2P Network. In *WWW*, 2002.
[8] C. P. Costa and J. M. Almeida. Reputation Systems for Fighting Pollution in Peer-to-Peer File Sharing Systems. In *Peer-to-Peer Computing*, pages 53–60, 2007.
[9] C. P. Costa, V. Soares, J. M. Almeida, and V. Almeida. Fighting pollution dissemination in peer-to-peer networks. In *SAC*, 2007.
[10] N. Curtis, R. Safavi-Naini, and W. Susilo. X^2Rep: Enhanced Trust Semantics for the XRep Protocol. In *ACNS*, pages 205–219, 2004.
[11] E. Damiani, S. D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *ACM Conference on Computer and Communications Security*, pages 207–216, 2002.
[12] J. R. Douceur. The Sybil Attack. In *IPTPS*, 2002.
[13] S. Garriss, M. Kaminsky, M. J. Freedman, B. Karp, D. Mazieres, and H. Yu. RE: Reliable Email. In *NSDI*, 2006.
[14] P. Gauthier, B. Bershad, and S. D. Gribble. Dealing with Cheaters in Anonymous Peer-to-PPer Networks. In *Technical Report of University of Washington*, 2004.
[15] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The Eigentrust algorithm for reputation management in P2P networks. In *WWW*, 2003.
[16] J. M. Kleinberg. The small-world phenomenon: an algorithm perspective. In *STOC*, 2000.
[17] L. Lamport, R. E. Shostak, and M. C. Pease. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, 1982.
[18] J. Liang, R. Kumar, Y. Xi, and K. W. Ross. Pollution in P2P file sharing systems. In *INFOCOM*, 2005.
[19] J. Liang, N. Naoumov, and K. W. Ross. Efficient Blacklisting and Pollution-Level Estimation in P2P File-Sharing Systems. In *AINTEC*, 2005.
[20] A. Mislove, M. Marcon, P. K. Gummadi, P. Druschel, and B. Bhattacharjee. Measurement and analysis of online social networks. In *Internet Measurement Comference*, 2007.
[21] B. Parno, D. Wendlandt, E. Shi, A. Perrig, B. M. Maggs, and Y.-C. Hu. Portcullis: protecting connection setup from denial-of-capability attacks. In *SIGCOMM*, pages 289–300, 2007.
[22] J. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. Epema, M. Reinders, M. van Steen, and H. Sips. Tribler: A Social-based Peer-to-Peer System. In *IPTPS*, 2006.
[23] P. Resnick, K. Kuwabara, R. Zeckhauser, and E. Friedman. Reputation systems. In *Communications of the ACM*, 2000.
[24] H. Rowaihy, W. Enck, P. McDaniel, and T. L. Porta. Limiting Sybil Attacks in Structured P2P Networks. In *INFOCOM*, pages 2596–2600, 2007.
[25] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-Resilient Online Content Voting. In *NSDI*, 2009.
[26] K. Walsh and E. G. Sirer. Fighting Peer-to-Peer SPAM and Decoys with Object Reputation. In *Workshop o the Economics of Peer-to-Peer Systems*, 2005.
[27] K. Walsh and E. G. Sirer. Experience with an Object Reputation System for Peer-to-Peer Filesharing. In *NSDI*, 2006.
[28] L. Xiong and L. Liu. PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities. *IEEE Trans. Knowl. Data Eng.*, 2004.
[29] H. Yu, P. B. Gibbons, M. Kaminsky, and F. Xiao. SybilLimit: A Near-Optimal Social Network Defense against Sybil Attacks. In *IEEE Symposium on Security and Privacy*, pages 3–17, 2008.
[30] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: defending against sybil attacks via social networks. In *SIGCOMM*, pages 267–278, 2006.