SpamResist: Making Peer-to-Peer Tagging Systems Robust to Spam

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

***Abstract*—Tagging systems are known to be particularly vulnerable to *tag spam*. Due to the self-organization and self-maintenance nature of Peer-to-Peer (P2P) overlay networks, the users in the P2P tagging systems are more vulnerable to tag spam than the centralized ones. In this paper, we propose SpamResist, a novel social reliability-based mechanism. For each tag search, SpamResist client groups the search respondents into two categories, namely *unfamiliar peers* and *interacted peers* according to the fact whether the client has interacted with such respondents. In addition, for the two different categories of peers, the client computes their *reliability degrees*, and then utilizes these reliability degrees as the weights to rank the search results. To obtain higher quality search results, we propose a *socially-enhanced mechanism*, considering social friends can share their previous experience and help improve both the performance and convergence of SpamResist. Finally, the experimental results illustrate that SpamResist can effectively defend against tag spam and work better than the existing search models in P2P tagging systems.**

I. Introduction

With the rapidly growing popularity of tagging services on the global Internet, people can share and tag different categories of resources, for example, photos in Flickr [1], URLs in Del.icio.us [2], videos in YouTube [3], papers in CiteUlike [4], and people in Fringe [5]. However, as shown in [7], it is widely accepted that there are some drawbacks in centralized tagging systems, such as the limited resources allocation and the vulnerability to Denial-of-Service (DoS) attacks. Based on the huge amount of available resources and without the single point of failure, Peer-to-Peer (P2P) tagging systems were proposed to address the above drawbacks. For instance, the open source project *Tagster* [6] developed a distributed hash table (DHT) application based P2P tagging system recently [7].

For a typical P2P tagging system, each specific *file* (such as photo and video) is annotated with some *tags*, and the relation $\langle file, tag \rangle$ that assigns a tag to a file is called an *annotation*, which maintains the association between the file and tag. When *client* issues a tag search to the system, the peer who annotated some local files with this tag will respond the client with these files. This peer is called *respondent*, and the files are called *response resources*. Moreover, a distributed index structure ensures that *tagging metadata* is always available to all the other participants even if not all peers are permanently online. Normally, the behaviors that peer A *downloads* some files from peer B are called *interactions* between A and B.

Many recent studies indicated that the centralized tagging systems are vulnerable to *tag spam*: erroneous or misleading tags that are generated by some attackers to confuse the normal users [8], [9]. Due to the self-organization and self-maintenance nature of P2P overlay networks, the participates in P2P-based tagging systems are more vulnerable to the tag spam than the centralized ones. For instance, some malicious users may repeatedly annotate some videos or photos with the erroneous tag, so that the normal good users without sufficient knowledge about other users may be misled to download an undesirable movie or photo.

Generally, the trusted moderators could be employed to identify good and spam tags for any resource in the tagging systems. However, due to the lack of centralized trusted authorities in common P2P tagging systems, the applicability of such moderators is questionable. In this paper, we propose SpamResist — a novel social reliability-based mechanism against tag spam for P2P tagging systems. In order to obtain quality tag search results, SpamResist encompasses two key mechanisms: *reliability mechanism* and *socially-enhanced mechanism*:

- *Reliability Mechanism:* For each tag search, client calculates a *reliability degree* for each respondent, and uses weighted averaging to compute the rank of the search results. Here, the computed weights are the reliability degrees of the owners of each response resource. Reliability degree is a personalized score assigned to each peer in the system by the client, and SpamResist proposes two algorithms for the client to calculate the reliability degrees of two categories of peers in the system respectively: *unfamiliar peers* and *interacted peers*. Unfamiliar peers denote the peers that the client has never interacted with, and interacted peers mean the peers that the client has interacted with before. For the unfamiliar peer, the client calculates his reliability degree with the statistical correlation of annotations between the client and the peer. For the interacted peer, SpamResist proposes *experience vector* to help the client store the latest n experiences

with each interacted peer, and the client can obtain the reliability degree of the interacted peer by averagely computing the elements of the experience vector with respect to the interacted peer. Because the tag search results presented to the client are ranked by the weighted averaging of reliability degrees of search respondents, the results provided by the *spammers* will be degraded to the end of the client's search result pages.

- *Socially-enhanced Mechanism:* To address some practical issues and enhance the reliability mechanism, we introduce the social networks to the P2P tagging systems. In socially-enhanced mechanism, the client may have many friends, and these friends are either acquaintances in reality or those online friends recognized in social networks. Thus, these reliable companions can provide many references based on their previous experiences to enhance the performance of SpamResist.

We conducted experiments with different configurations, and compared SpamResist with the existing tag search models: *Boolean* [10], *Occurrence* [11], *Coincidence* [9], and *PINTS* [7]. In order to validate the effectiveness of SpamResist, we did not only compare SpamResist with PINTS, the current search model of the P2P tagging systems, but also implemented the representative search models of the centralized tagging systems — Boolean, Occurrence and Coincidence — in the P2P infrastructure for comparison. The evaluation results show that SpamResist can defend against tag spam from various attackers more effectively than the existing search models in the P2P-based tagging systems.

To the best of our knowledge, there is not any existing study on defending against tag spam in P2P tagging systems. Our research contributions are as follows:

- We propose SpamResist, a novel social reliability-based mechanism against tag spam in the P2P tagging systems.
- We are the first to introduce the social network to P2P tagging systems.
- We compare SpamResist with various search models under different threat models, and present the effectiveness of SpamResist against tag spam.

The rest of this paper is organized as follows. We present the related work in Section II. Section III describes the details of SpamResist, and presents some examples to illustrate the principle of SpamResist clearly. Then, the simulation methodology and evaluation results are discussed in Section IV. Finally, we present conclusion in Section V.

II. Related Work

A. P2P Tagging System

As a new infrastructure of tagging services, the existing work on the P2P tagging systems has received little attention so far. To our knowledge, Tagster [6], [12] is the only one P2P tagging system in the practical application. The study in [7] reported PINTS, the approach adopted in Tagster which could address the problems of information aggregation and utilization in the P2P-based tagging environment.

PINTS proposes *feature vector* for each client to store a characteristic score of each peer in the system, and the score is computed by the similarity frequency of the resources between the client and the peer. Using the feature vector, PINTS predicates the possible respondents in the network, and sends tag search request to them. Tagster treats the search results returned by these possible respondents as the approximate global tagging information in the system. The experiments in [7] proved that PINTS could obtain a relatively accurate global tagging information with low communication overhead.

However, the design of PINTS did not consider how to defend against spam. To the best of our knowledge, there is not any existing study on defending against tag spam in P2P tagging systems.

B. Defense Mechanisms against Tag Spam

Currently, there are some mechanisms proposed to address tag spam in the centralized tagging systems. In general, these mechanisms can be grouped into three categories [8]: *detection-based* mechanisms, *interface-based* mechanisms and *demotion-based* mechanisms.

Detection-based Mechanisms: This category of mechanisms are mainly based on the principles of statistical analysis and machine learning. The study in [13] investigated the usefulness of different machine learning algorithms on the calculations of features. Besides, based on determining relevant features to tagging systems, the author transferred those approaches to identify those spammers. The studies in [14], [15] proposed the algorithms which could detect the tag spam through training and learning. Based on the notion that similar users and annotations tend to use the same language, the study in [16] introduced *language model* to address the problem of spam in tagging systems.

Interface-based Mechanisms: These methods attempt to either hide or restrict access to actions that make users contribute content. CAPTCHAs [17] can also be used to prevent automated account creation or automated tag spam annotation.

Demotion-based Mechanisms: In this category of anti-spam mechanisms, the algorithms can return the most popular list and degrade the rank of spam to the end of search results. The study in [18] took into account spam by proposing a credibility score for each user based on the quality of the tags contributed by the user. Studies [8], [9] proposed a simplified model of tagging behavior in tagging system and compared different ranking methods for tag-based searching.

III. Design Rationale of SpamResist

In this section, we describe two key mechanisms of SpamResist and the solution to the encountered practical issue. First, we describe *reliability mechanism* in Section III-A, and the *socially-enhanced mechanism* in Section III-B. We then present the solution to the encountered practical issue in Section III-C.

A. Reliability Mechanism

In SpamResist, if the client wants to acquire a specific file F, he will issue a tag query t to the network. Each peer who annotated his own files with t will respond the search request. After receiving the responses from network, the client aggregates these results into some *search result pages*. In the result pages, each result can be ranked by the weighted averaging of reliability degrees of its owners. Thus, the ranking score of each result may be interpreted as a personalized estimate of the authenticity of the annotation $\langle F, t\rangle$, thus help users make an informed decision to download or disregard the file.

In SpamResist, client A assigns a personal reliability degree to each other participant in the system. Specifically, client A uses $R_{A,B}$ to denote the reliability degree of peer B from A's perspective. Generally, there are two categories of peers in the system for the client: *unfamiliar peers*, the peers that the client has never interacted with, and *interacted peers*, the peers with whom the client has interacted before. Reliability mechanism provides two methods for the client to compute the reliability degrees of the above two categories of peers respectively.

Unfamiliar peers. To obtain the reasonable reliability degrees of the unfamiliar peers, we propose the correlation technique to compute the reliability degrees. Specifically, for the unfamiliar peer B, the client A does not have any referential experience to calculate the reliability degree to B. Using the correlation technique, the client A computes the correlation of annotations between A and B, and treats the correlation as the reliability degree, $R_{A,B}$, with respect to B. The equation that computes the correlation between the client A and the peer B is described as follows:

$$R_{A,B} = \frac{\sum_{f_i \in F} \left(\sum_{t_j \in C_{f_i}} |P(f_i, t_j)| \right)^2}{\sqrt{\sum_{f_i \in F} \left(\sum_{t_j \in T_A} |P(f_i, t_j)| \right)^2} \sqrt{\sum_{f_i \in F} \left(\sum_{t_j \in T_B} |P(f_i, t_j)| \right)^2}} \tag{1}$$

where

- F: The set of files owned by A and B in common.
- f_i: The i^{th} file of the common file set F.
- C_{f_i}: The set of the tags annotated by A and B in common to the file f_i.
- t_j: The j^{th} tag of the tag set.
- T_x: The set of tags annotated by the peer x to the file f_i.
- $P(f_i, t_j)$: The set of annotation that annotated f_i with t_j.
- $|P(f_i, t_j)|$: The size of $P(f_i, t_j)$.

In this case, the range of $R_{A,B}$ is $[0, 1]$. Higher value indicates that the peer B is more reliable for A, and small $R_{A,B}$ indicates either the peer B is unreliable for A or the lack of any significant relationship between the client and the peer.

We also notice that the client can only compute accurate and significant reliability degrees for others if he has himself cast a sufficient high ratio of the *consistent* annotations. Consistent annotations mean that both the files and the tags of two annotations are the same. This restriction provides a strong incentive for peers to participate in annotating, since the peers that do not annotate correctly and actively will find the quality of the estimate they compute noticeably degraded. Indeed, a client can still benefit from SpamResist by annotating honestly but inactively, suppressing the sharing and dissemination of erroneous tags to the system.

Interacted peers. For the peers that the client has some experiences with, SpamResist calculates their reliability degrees based on the client's previous experiences with them. SpamResist proposes *experience vector*: each client stores the previous experiences from the interacted peers in his own experience vector. Specifically, for the peer B that client A has interacted with, A maintains a vector of length n storing the most recent n experiences with B, and as new experiences are appended the oldest ones are removed. The experience vector, $EV_{A,B}$, is $\langle \vartheta_{A,B,1}, \vartheta_{A,B,2}, ..., \vartheta_{A,B,n} \rangle$ where $\vartheta_{A,B,k}$, $k = 1, ..., n$ are values between -1 to 1. The calculation of $\vartheta_{A,B,k}$ is based on the evaluation, $\lambda_{A,B,k}$, of the interaction represented by the kth element of $EV_{A,B}$, as well as the reliability degree, $R_{A,B,k}$, from A to B at the interaction represented by the kth element of $EV_{A,B}$. The specific equation is described as follows:

$$\vartheta_{A,B,k} = \lambda_{A,B,k} \cdot R_{A,B,k}, \qquad 1 \le k \le n \tag{2}$$

where

- $\lambda_{A,B,k}$: The evaluation of the interaction which is represented by the kth element of $EV_{A,B}$, and the value is -1 or $+1$, where -1 represents erroneous annotation and $+1$ represents accurate annotation.
- $R_{A,B,k}$: The reliability degree, from A to B, of the interaction which is represented by the kth element of $EV_{A,B}$.

The client A can obtain the reliability degree with respect to the interacted peer B through calculating the average value of B's experience vector. Moreover, each client can set a parameter R_{min} as the minimum reliability that a peer can be considered trustworthy by the client. Client will not respond to the peers which are currently considered not trustworthy. Due to the disregard of the tag query, SpamResist provides another incentive to enhance the honest annotating and sharing of peers in the system.

B. Socially-enhanced SpamResist

In the practical applications, one representative issue, which can not be addressed by the above reliability mechanism, is the lack of experience. For instance, due to the absence of the common files with unfamiliar peer B, the client A is unable to compute the ideal reliability degree, $R_{A,B}$, using the reliability mechanism. Another example, due to the insufficient experience, a newcomer probably does not have the sufficient annotations to compute the accurate reliability degrees with

respect to most peers in the system, thus he would be a victim. To address the above problem, we explore the solution through utilizing the characteristic of tagging systems. The study in [19] indicated that the social network enables not only efficient, reliable resource discovery and recommendation with a low additional overhead, but also a significant improvement on the performance of the P2P system. Thus, in order to construct socially-enhanced SpamResist, we introduce *friend-based scheme* to P2P tagging system.

Friend relationship establishment. In socially-enhanced SpamResist, the client may have many friends and store their information in the client's *friend list*. Any newcomer can be invited by a participating peer, and thus added into the system; meanwhile the inviter will become the friend of the newcomer automatically. The invitation mechanism ensures that client at least owns one friend in the system. Besides the method of invitation, we can establish friend relationships with the peers who are our acquaintances in reality or those online friends recognized in social networks. SpamResist introduces the friend scheme based on the fundamental fact that the friends are more reliable than those anonymous peers in P2P tagging systems. Thus, the client could set the reliability degrees of his friends to 1 directly. Note that, the client's friends may be malicious or compromised, thus we will present an approach to addresses this practical issue, in Section III-C.

Enhanced mechanism. Due to the introduction of friend-based scheme, the client can utilize these reliable companions to address the practical issue mentioned above with an approach which is similar with the *majority voting*. After the client ranks his search results, he will find out all the files which have the ranking scores lower than 0.5 in his ranking results, and then the client sends all his friends some *enhanced requests* which demand these friends to compute ranking scores for those files. For a result, if more than half of the friends return the ranking scores of the above file higher than 0.5, SpamResist will re-locate the position of the file in the client's result page according to the new ranking score by computing the average of these friends' returned scores higher than 0.5. Otherwise, the client maintains the original rank unchanged.

For example, file $a.jpg$ has the ranking score 0.4 in the client's result page. Using socially-enhanced scheme, the client gets ranking scores ,0.7, 0.8 and 0.4, from his three friends, so he should replace $a.jpg$'s original ranking score 0.4 with $(0.7 + 0.8)/2 = 0.75$, so that the position of $a.jpg$ in the client's result page is changed. Another example, the ranking scores from the client's friends are 0.5, 0.3 and 0.2 respectively. Because lower than half of the friends provide the ranking scores higher than 0.5, the ranking score of the file $a.jpg$ remains unchanged.

The design of enhanced mechanism is based on the following considerations. The fact that file has a low ranking score in the client's result pages is probably caused by the reason that the client has no sufficient experiences with the owners of the file. Thus, the client chooses to seek help from his friends to compute the ranking score of the file again. If the newly computed ranking score is higher than 0.5, we consider this value a relatively reliable score, so the client will re-locate the position of the file in his result page; otherwise, the client will maintain the position of the file unchanged.

C. Practical Issue

In the practical applications, some friends may be online deceivers or compromised. To avoid the harms incurred by malicious friends, SpamResist provides the friends' reliability degrees, $R_{A,f}$, for helping the client find out these unreliable friends. When the client evaluates -1 to a search result, besides computing the reliability degrees in experience vectors for result providers, SpamResist also examines whether the client's friends also share this erroneous result. If there are indeed some friends providing this erroneous result, their reliability degrees will be decreased. Once the reliability degree of the client's friend drops to below R_{min}, the client will not send the enhanced request used in the socially-enhanced mechanism to this friend. Thus, the malicious friend can not continue to provide the erroneous information. Meanwhile, if the client casts $+1$ to a search result, SpamResist also examines whether some client's friends annotate the result with the current tag. This mechanism ensures the friends who have the low reliability degrees can recover their reliability degree. The specific algorithm is as follows:

$$R_{A,f} = \begin{cases} \max(-1, R_{A,f} - \beta n^2) & \text{if result is erroneous} \\ \min(1, R_{A,f} + \alpha) & \text{otherwise} \end{cases} \tag{3}$$

where

- n: the number of consecutive discoveries of erroneous annotations from friend f (including the last one).
- β: the penalty factor given to friend f for each erroneous annotation result evaluation.
- α: the recompense given to friend f for each correct annotation result evaluation.

Note that, in this case, the reliability degree from A to f decreases faster than it increases. Aiming at severely penalizing malicious friends, SpamResist weights β by the square of the number of erroneous results discovery. Moreover, SpamResist uses different penalty and recompense factors, and we propose to set $\beta > \alpha$.

IV. Evaluation

To evaluate the performance of SpamResist, we developed a prototype of P2P tagging system based on Chord [20]. Note that the focus of our experiments is the impact of tag spam on the search result, but not the specific method on the aggregation of tagging information.

In order to illustrate the effectiveness of SpamResist, we should compare SpamResist with the existing search models in the P2P tagging systems. To the best of our knowledge, PINTS is the only search model used in the current P2P tagging

systems. Although no other mechanism against tag spam in P2P tagging systems has been proposed, there are a few models that focus on defending against tag spam in centralized tagging systems. To provide a more meaningful evidence to present the performance of SpamResist, we need to consider the following question: *If we implement the search models used by the centralized tagging systems into P2P tagging systems, can SpamResist work better than those models on defending against tag spam?*

Bearing these in mind, besides PINTS, we also implemented three representative search models, which are formerly used in the centralized tagging systems, in our P2P tagging system including Boolean [10], Occurrence [11], and Coincidence [9].

A. Simulation Setup

Our prototype of P2P tagging system consists of the following main models:

Network model. Because our focus is the dissemination of tag and file in the network, we assume the perfect overlay routing and tag querying in the following experiments. Moreover, the tags and files of each peer are always shared, and the transfer time is assumed to be negligible since our focus is the robustness against spam.

Peer model. The network is composed of P peers including good and malicious peers. We utilize Kleinberg model [21], a widely accepted social network model, to establish the friend relationships for all the participants in the system. We then assign 12 friends to each peer averagely according to the measurement results in [22]. At the startup of simulation, each good peer annotates all his files with the correct tags, and then they search the tags and download some of them according to the ranking score of each search result. If a good peer downloads a file and finds the one not match the tag he searched, he may delete this file or annotate it with the correct tags and publish these annotations. In our experiments, we assume the good peers evaluate each download actively. Besides, a malicious peer shares the misleading annotations and participates in the overlay network to spread them — this attempt to undermine the performance of the system.

File model. There are F files in the system. Because our focus is the tag spam in the search result, we do not consider the concept of *file version*[1]. At the startup of simulation, file selection follows Zipf distribution with the parameter $\alpha = 0.8$ [24]. We assume there are T size of vocabulary for peers to tag their files, and their selections of annotations follow the distributions measured in [25].

B. Search Models

This section, we describe four tag search models which are used to compare with SpamResist: Boolean, Occurrence, Coincidence, and PINTS. We describe the search strategies of the four models, and discuss how they work in the P2P tagging systems.

[1]The definitions of the term title and version can be found in [23]

Boolean model. Boolean model is one of the representative search models in the centralized tagging systems (e.g., Slideshare [10]). The search strategy of Boolean is that the system randomly ranks the results associated with the search tag.

Occurrence model. For Occurrence model, a search model in the centralized tagging systems (e.g., Rawsugar [11]), the system ranks the search results based on the number of annotations containing the searched tag, and returns the top ranking results.

Coincidence model. Coincidence model assigns each user in the centralized tagging systems a score computed by the number of the annotations $\langle file, tag \rangle$ overlapped with other users in the system [8], [9].

Because the above three models are all used in the centralized tagging systems, in order to compare with SpamResist, we discuss how to implement them in the practical P2P tagging systems. Currently, the study in [7] has provided an approach which can aggregate the decentralized information of annotations with feature vector model in the P2P tagging systems. Tagster [6], a practical application of the P2P tagging system, has proved the applicability of feature vector model. There are also some other methods that can be used to obtain the global tagging information in the P2P overlay, such as gossip protocol and flooding mechanism. In our experiments, we adopted the gossip protocol to obtain the global tagging information.

PINTS model. Until now, PINTS [7] is the only search model in the P2P tagging systems, and it has been developed in Tagster [6]. The search strategy of PINTS is that the client first generates a feature vector to store a characteristic score (the algorithm mentioned in Section II-A) for each peer in the system; then, using the feature vector, the client aggregates the information of annotations selectively, and randomly ranks the search result.

C. Threat Models

In this section, we describe three threat models existing in practical tagging systems.

Random attack model. For the random attackers, they always select some of their files and annotate them with some erroneous tags randomly, in order to mislead those normal participants in the systems. Normally, the random attack acts independently, that is, these bad peers are "lousy taggers".

Targeted attack model. In some cases, malicious peers can collude and mount some intelligent attacks. These colluding peers first share some particular files owned by them in common. Then, they annotate these files with misleading tags to make these files easy to be searched by the other good users.

Disguised attack model: Some malicious peers may launch a trickish attack. They annotate local files with both the correct and the erroneous tags, then publish these annotations to the network. The existing anti-spam mechanisms (e.g.,

TABLE I: Default Parameters in Experiments

Parameter	Value	
Number of Files F	10,000	
Size of Vocabulary T	5,000	
Number of Files in Result R	10	
Penalty Factor β	0.2	
Recompense Factor α	0.1	
Parameter	**Good Peers**	**Attackers**
Number of Peers	800	200
Owned Files	30	50
Search and Save Rate	0-5 / day	0

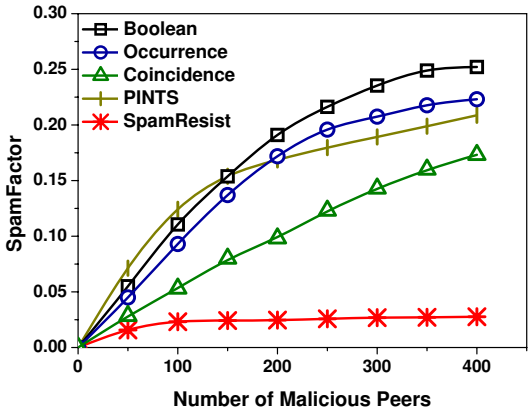

Fig. 1: Impact of the Number of Random Attackers

Coincidence) will be a victim when encountering this trickish attack.

D. Performance Metric

This section, we discuss the metric for evaluating our experiments. Because our purpose is to evaluate the impact of tag spam on the search result, we make use of *SpamFactor* [8], [9], a metric accepted widely to quantify the "spam impact" on search result. SpamFactor is affected by both the number of spam files and their position in the search result. The higher the position of a bad file in the result page, the higher SpamFactor. Furthermore, the study in [9] has argued that "SpamFactor less than 0.1 is 'tolerable' in the sense that the spam files will be few and towards the bottom of the result page".

E. Evaluation Results

We run experiments with various configurations. Table I summarizes all the parameters and their default values in the following discussions.

Impact of the number of random attackers. While keeping the number of total peers in the system as $1,000$ constantly, we vary the number of random attackers from 0 to 400. The result shown in Fig. 1 clearly indicates that Boolean, Occurrence and PINTS all suffer from the random attacks when the percentage of malicious peers grows higher than 10%. The reason that the SpamFactors of three models increase above 0.1 so quickly is that their designs do not take spam problem into account. We

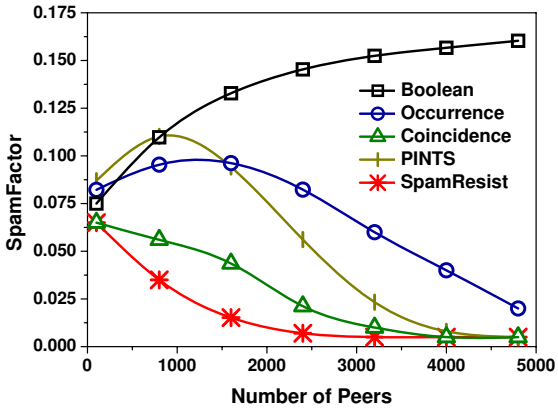

Fig. 2: Impact of the Number of Peers

observe that Coincidence model can work better than the above three models; however, after increasing the percentage of random attackers above 20%, the SpamFactor of Coincidence model is higher than 0.1 — the "tolerable" ratio. This is because that, as the number of attackers proliferates, the probability of coincident annotations from attackers also becomes higher. Thus, we consider that Coincidence model can not work well in the P2P tagging system with the percentage of random attackers higher than 20%. For SpamResist, even if the percentage of random attackers is more than 40% in the system, it can still maintain the SpamFactor lower than 0.05.

Impact of the number of peers. While increasing the number of total participants in the system, we keep the percentage of random attackers 10% constantly. The result shown in Fig. 2 presents that, as the increase of the number of peers in the system, only the Boolean corresponds to the SpamFactor higher than 0.1. This is because that the random attackers can not launch the attack to some particular tags collusively. Note that the initial degradation of Occurrence and PINTS are due to the lack of enough peers in system to generate the referential annotations. Furthermore, we also notice that the SpamFactor of SpamResist can quickly drop to 0.025 when the number of total peers is more than $1,000$.

Impact of the random attacks. In this experiment, we compare SpamResist with the existing search models on both the performance and the convergence under the environment with 20% random attackers. The result shown in Fig. 3 presents that the SpamFactor of Boolean model always maintains higher than 0.15 during the experiment. Because there is a comparatively large percentage of attackers, the occurrence frequency of erroneous annotations is high. Thus, the SpamFactor of Occurrence decreases to below 0.1 in 8 days. The SpamFactors of Coincidence model and PINTS model can converge to below 0.1 in 3 and 5 days respectively. SpamResist performs the best convergence because its SpamFactor can decrease to below 0.1 in only 2 days. This is due to the socially-enhanced mechanism of SpamResist.

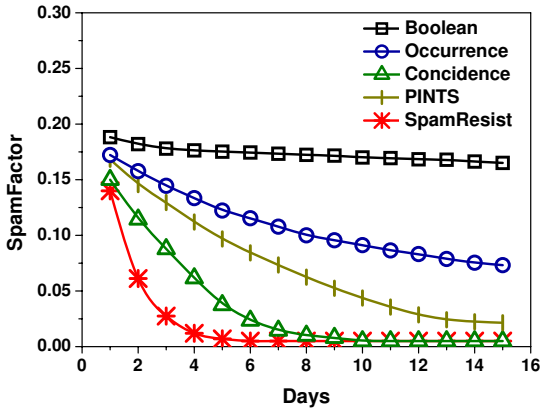

Fig. 3: Impact of the Random Attacks

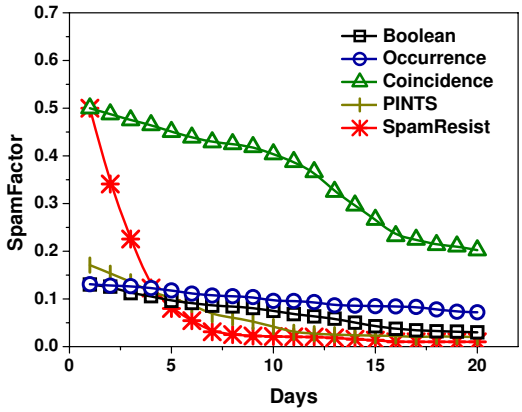

Fig. 5: Impact of the Disguised Attacks

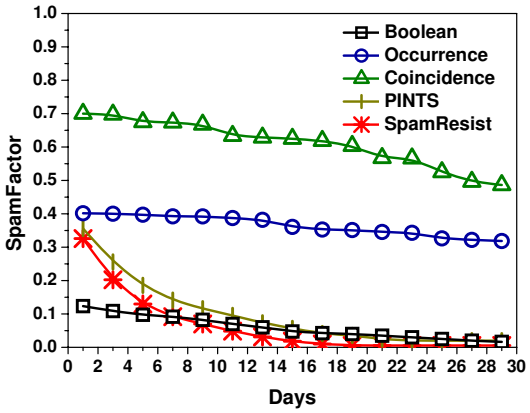

Fig. 4: Impact of the Targeted Attacks

Impact of the targeted attacks. In this experiment, we compare SpamResist with the existing search models on defending against the targeted attackers in P2P tagging system. The result shown in Fig. 4 presents that the SpamFactor of Boolean model decreases gradually and becomes lower than 0.1 after 4 days. This is because that the mechanism of Boolean model is to choose the annotations randomly, the search results of Boolean contain spam with a low ratio. Thus, Boolean model does not suffer from the targeted attack significantly. However, comparing with Boolean model, the SpamFactors of Occurrence and Coincidence are always upon 0.35 and 0.5, respectively, during the experiment. For Occurrence model, because the number of the misleading annotations are promoted, the model returns large erroneous results; for Coincidence, because the coincidence factors of the targeted attackers are boosted, all erroneous annotations are promoted in search results. Thus, this experiment proves that Occurrence and Coincidence can not work well when encountering the targeted attack. We observe that both the SpamFactors of PINTS and SpamResist can converge to below 0.1 in 11 days and 5 days respectively, and the SpamFactor of SpamResist reaches the lowest SpamFactor among all the models after only 7 days.

Impact of the disguised attacks. In this experiment, we compare SpamResist with the existing search models on defending against the disguised attackers in the system. The result shown in Fig. 5 presents that Boolean model and Occurrence model can work well when encountering the disguised attack, and their SpamFactors can drop to the values lower than 0.1 after 5 and 9 days respectively. We notice that Coincidence model returns the high SpamFactor during this experiment. The reason is that, due to the lack of the adequate interacted experiences, the client treats those disguised attackers as coincident peers. Thus, the SpamFactor of Coincidence is high. PINTS model has a good performance under the attacks of disguised peers, and its SpamFactor drops to lower than 0.1 after 6 days. Besides, at the startup of the experiment, the SpamFactor of SpamResist is very high; however, SpamResist can converge quickly and decrease its own SpamFactor to below 0.1 in only 5 days. This is due to both the interaction-based evaluation and the socially-enhanced mechanism of SpamResist.

V. Conclusion

Current Peer-to-Peer (P2P) tagging systems are highly vulnerable to the tag spam. SpamResist is a novel social reliability-based mechanism on defending against the tag spam in P2P tagging systems. In order to obtain the quality search results, SpamResist includes two key mechanisms: reliability mechanism and socially-enhanced mechanism. After each tag search, the client can compute the reliability degrees of two categories of respondents, unfamiliar peers and interacted peers, by the reliability mechanism. Then, the client makes use of these reliability degrees as the weights to rank the search results. Moreover, SpamResist provides the friend-based scheme to enhance both the performance and convergence of defending against tag spam. As shown in our experimental results, SpamResist can work better than those

existing search models on defending against tag spam in P2P tagging systems.

REFERENCES

[1] "Flickr, http://www.flickr.com/."
[2] "Del.icio.us, http://del.icio.us/."
[3] "Youtube, http://www.youtube.com/."
[4] "CiteUlike, http://www.citeulike.org/."
[5] S. Farrell and T. Lau, "Fringe contacts: People tagging for the enterprise," in *Collaborative Web Tagging Workshop in conjunction with WWW*, 2006.
[6] "Tagster, http://isweb.uni-koblenz.de/research/tagster."
[7] O. Görlitz, S. Sizov, and S. Staab, "PINTS: Peer-to-peer infrastructure for tagging systems," in *IPTPS*, 2008.
[8] P. Heymann, G. Koutrika, and H. Garcia-Molina, "Fighting spam on social web sites: A survey of approaches and future challenges," *IEEE Internet Computing*, vol. 11, no. 6, pp. 36–45, 2007.
[9] G. Koutrika, F. A. Effendi, Z. Gyöngyi, P. Heymann, and H. Garcia-Molina, "Combating spam in tagging systems," in *AIRWeb*, 2007.
[10] "Slideshare, http://slideshare.net/."
[11] "Rawsugar, http://rawsugar.com/."
[12] O. Görlitz, S. Sizov, and S. Staab, "Tagster - tagging-based distributed content sharing," in *ESWC*, 2008, pp. 807–811.
[13] B. Krause, C. Schmitz, A. Hotho, and G. Stumme, "The anti-social tagger: detecting spam in social bookmarking systems," in *AIRWeb*, 2008, pp. 61–68.
[14] Z. Kyriakopoulou and T. Kalamboukis, "Combining clustering with classification for spam detection in social bookmarking systems," in *RSDC*, 2008.
[15] A. Gkanogiannis and T. Kalamboukis, "A novel supervised learning algorithm and its use for spam detection in social bookmarking systems," in *RSDC*, 2008.
[16] T. Bogers and A. Bosch, "Using language models for spam detection in social bookmarking systems," in *RSDC*, 2008.
[17] L. von Ahn, M. Blum, N. J. Hopper, and J. Langford, "Captcha: Using hard ai problems for security," in *EUROCRYPT*, 2003, pp. 294–311.
[18] Z. Xu, Y. Fu, J. Mao, and D. Su, "Towards the semantic web: Collaborative tag suggestions," in *Collaborative Web Tagging Workshop in conjunction with WWW*, 2006.
[19] 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.
[20] I. Stoica, R. Morris, D. Liben-Nowell, D. R. Karger, M. F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Trans. Netw.*, 2003.
[21] J. M. Kleinberg, "The small-world phenomenon: an algorithm perspective," in *STOC*, 2000.
[22] A. Mislove, M. Marcon, P. K. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and analysis of online social networks," in *Internet Measurement Comference*, 2007, pp. 29–42.
[23] J. Liang, N. Naoumov, and K. W. Ross, "Efficient blacklisting and pollution-level estimation in p2p file-sharing systems," in *AINTEC*, 2005.
[24] J. Liang, R. Kumar, Y. Xi, and K. W. Ross, "Pollution in p2p file sharing systems," in *INFOCOM*, 2005.
[25] R. Li, S. Bao, Y. Yu, B. Fei, and Z. Su, "Towards effective browsing of large scale social annotations," in *WWW*, 2007, pp. 943–952.