Adversarial information retrieval in distributed systems

Jelle Licht, Johan Pouwelse

***Abstract*—Trust is fundamental to any human interaction. Technology can assist in determining trustworthiness of information shared by strangers. A plentiful amount of techniques have been developed to do exactly this. However, many existing systems delegate ultimate trust to a predatory company, or assume that there is some out of band information available on which to base initial estimates of trustworthiness. A system without these drawbacks that consistently classifies peers as honest or dishonest does not currently exist. The goal of this paper is to evaluate and analyze the existing systems and their shortcomings, as well as look at ongoing efforts to make use of the properties of distributed ledger technology for a way forward.**

I. Introduction

Human interactions are based on trust between individuals. The willingness to trust others can be diminished by continuous dealings with incompetent or abusive behavior, and technological systems are not exempt from this. Indeed, whereas advances in predictive models and behavioral analytics can advise us regarding the trustworthiness of certain individuals, the value judgment of whether to trust is still a essentially human one. This problem is exacerbated when assessing the trustworthiness of hundreds, thousands or even millions of individuals, as is often the case in decentralized content sharing systems.

These systems rely on trust for their core operations once the amount of content is intractable for an individual to verify and/or authenticate. Some sort of gatekeeping or filtering is required. For systems like FaceBook, this responsibility is delegated to a central authority. A drawback of allowing FaceBook, or any other central authority for that matter, to make these decisions is that it can lead to inadvertently taking part in a emotion study [1]. A general lack of transparency on the inner workings of such a system [2] make it difficult to identify biases in the selection of content as presented. A decentralized and transparent solution is needed with accountability and trust built-in. More specifically, we take a view that focuses on dealing with spam in these systems.

This work means to provide insight into the existing designs and techniques in this area, as well as provide a framework with which to analyze and compare the shortcomings of existing and proposed solutions. In this paper we demonstrate the various ways in which existing systems deal with dishonest peers, specifically by making use of votes cast by peers. Our major contributions are as follows:

jlicht@fsfe.org, j.a.pouwelse@ewi.tudelft.nl

[1] https://www.theguardian.com/technology/2014/jun/30/facebook-news-feed-filters-emotion-study

[2] http://raley.english.ucsb.edu/wp-content/Engl800/Pasquale-blackbox.pdf

- A survey of different dealings with the potential maliciousness of peer-supplied information in decentralized systems.
- A case study showing how a real-world file-sharing systems makes use of votes cast by users to improve relevance of search results.

The rest of this paper is organized as follows: Section II introduces the problems as tackled by this paper, as well describing the general challenges plaguing decentralized systems. Section III describes mitigation strategies which apply only in specific cases and a more general approach to building trust. Section IV regards a case study using Tribler, a real-world fully distributed content sharing and streaming system Section V gives the concluding remarks of this paper.

II. Problem description

It is hard to divide the world in good and evil, even more so for a machine. When exchanging information with a multitude of individuals, the distinction has to be made between honest and dishonest peers in order to make informed decisions.

A key feature of decentralized systems is information sharing. In any but the most trivial systems, peers can not feasibly share all information available in the system: Peers need to use information retrieval techniques to find relevant information. A dishonest peer can influence the process of information retrieval depending on the specific system architecture. By gaining control of trusted parts of the system, one can choose to ignore, subvert or simply monitor peer interaction with the decentralized system. This can be effectively equivalent to denying certain or all peers service or threatening legal actions to users of such a system. Deciding which peers to

Decentralized system architectures were originally used and actively researched as a means to make systems more robust against censorship. Properly designed decentralized system exhibit attributes such as scalability, trustworthiness and reliability as well. Since 1999 the usage of p2p file-sharing systems has steadily increased, bringing with it an increased amount of interest in how these systems function. This has been a double edged sword, as interest come from both well-meaning users as well as more adversarial parties aiming to sabotage these systems. A modern decentralized file-sharing system needs. to take the motivations of adversaries and risks for legitimate users of the system into account. Tribler is one of such decentralized file-sharing systems that aims to provide users with a robust and censorship-proof service.

Modern examples of widely used decentralized systems

are the BitCoin blockchain [3], BitTorrent [4] and the Internet itself. Decentralization trends that have been ongoing since the inception of the Internet, combined with bursts of intense research activity have contributed to a sort of arms race between designers of decentralized systems and those perceiving harm by the successful development and deployment of decentralized systems. This has led to the creation increasingly complex decentralization schemes, while these adversaries have come up with social, legal and technical means to prevent designers and users of these systems from being successful in going about their business.

Interactions with honest peers can then be trusted, as well as the corollary of this statement Voting systems can alleviate some of the more serious weaknesses seen in most file-sharing systems. A potential issue with basing spam prevention measures on active user participation is that users might not be properly incentivized to act for the greater good of the system. In the most extreme cases, entities can try to shut down all services by launching a DDoS attack conducted either in the open or anonymously [11].

Most of the theoretical models rely on two central assumptions:

1) Cooperative users vote in a similar way, properly classifying spam vs authentic content.
2) Cooperative users vote.

Prior research challenge both of these assumptions in an empirical study [8]. As the first assumption can be seen as a stronger one than the second, disproving the second also allows the first to shown as optimistic in realistic scenarios.

As a corollary, this also means that cooperative users often help polluters spread misinformation and spam in the network without being aware of this. Any proposed solution for dividing the world in good and evil has to take into account that even honest peers can commit to abusive behavior because of ignorance.

We aim to give an overview of the abuse types adversaries have taken to prevent users from finding relevant content by spamming some part of the service.

A. Index poisoning and routing table poisoning

A relatively simple way of preventing a user from downloading certain content is making sure the user has no way of finding said content. As described in [19], index poisoning can take place when users depend on other, potentially malicious users for locating content. Index poisoning takes place when users have no way to distinguish content advertised by malicious vs cooperative users. Index poisoning is effective because an adversary only has to advertise non-existing or corrupt locations of content, after which a naive user will start the expensive process of following up on finding this advertised content.

If an adversary is able to advertise the misinformation in a superior way, this can lead to users repeatedly trying to make use of the corrupted index information before eventually stumbling on a correctly advertised piece of content by happenstance. By then, the damage is done, as most proper decentralized file-sharing systems rely on other users dealing with the same piece of content before being able to download it.

Another issue starts to rear its head when an adversary is able to contribute and sustain large enough of ostensibly cooperative peers to the network. As long as no malicious behavior is undertaken or detected, adversary-controlled nodes start to become entwined in the distributed routing tables of normal users. An adversary can employ this position of power to monitor traffic, or even deny services to any subset of users, or event deny services related to a specific piece of content [12].

A way to deal with index- and routing table poisoning is to routinely label misleading information, and purge it from local information stores. The advantages of this approach are two-fold. First of all, the peer will no longer make use of misleading information, and secondly the peer will no longer contribute to the problem by distributing the misleading information. The challenge then becomes to identify with reasonable certainty which pieces of information are misleading.

B. Content poisoning

An orthogonal method for the adversary to deny users of p2p networks services is to actively flood the network with mislabeled content. Compared to the index poisoning and routing table poisoning method, this method does actually lead to content being available on the network [10]. If the content has to be downloaded in its entirety before a user is able to determine its authenticity, this can quickly lead to entire swarms of peers downloading and in turn sharing spam. One way of dealing with content poisoning is by estimating the probability of the content being authentic.

To estimate the pollution of a file-sharing network, the authors of [10] differentiate between natural and intentional pollution, but conclude that natural pollution is usually limited to a negligible amount. A crawler collects metadata an availability information on the content in a certain network in a best-effort to create a snapshot.

C. Stream poisoning

The application of live p2p streaming of content gives an extra set of constraints which make it especially sensitive to stream poisoning. Dishonest peers can collude to periodically upload corrupt data to honest peers. Depending on how robust the transfers are, honest peers might have to re-download blocks, chunks or even entire files, thereby slowing down the network with unnecessary work. These same considerations also exist for non-live streaming systems, but in that case a system can already be considered usable if honest peers are able to use the system within some reasonable amount of time. On the other hand, even a two minute delay could already be considered too much for a live streaming system.

D. Tag spam

Tagging is the process of annotating content with a tag. P2p systems can benefit from tagging by providing a self-regulating

[3] https://www.bitcoin.com/
[4] http://www.bittorrent.com

fine-grained filter. When tags are properly applied to content, search performance should go up by more clearly describing and distinguishing the content available in a network.

E. votes-spam

If the assumption holds that colluders act in detectable patterns, techniques employed for identifying Web link farm spam pages can also be employed on partial views of a network [18], [5]

SumUp [15] is one the systems designed to be resilient against large swaths of colluding malicious users. It limits the amount of influence colluders have by introducing a bastardized version of the MaxFlow problem.

TorrentTrust [14] extends upon the Credence system by taking into account user trust. The authors also incorrectly state that Credence is by definition a centralized scheme with a centralized certificate issuer. This is arguably the case for the implementation of Credence, but [16] clearly states that any different gate-keeping scheme can be used.

III. Solutions

This section provides an overview of mitigation strategies to deter or limit the effect an adversary can have on the quality of service of the file-sharing system. Ways to prevent dishonest peers from being successful can be labeled in one of two types. The first one is to prevent or disincentivize an adversary from exhibiting the malicious behavior. The second type mitigates or isolates the effects of the malicious behavior such that honest peers can make use of the service unhindered.

Having defined the plethora of ways in which an decentralized file-sharing system can be abuses and secured in Section II, the decision has to be made on how to categorize gathered information according to trustworthiness.

Building trust among peers is a well researched topic, and one way of looking at the problem is to see interactions between peers as edges in a graph, with vertices denoting peers. Verifying an interaction implies trusting another peer to some extent. This way of defining trust in a network leads to Web of Trust. [6]

Credence [16] introduces the concept of object-based reputation as a way to estimate content authenticity. A simple voting protocol is used where a vote can be cryptographically traced back to the voter. In principle, an adversary can employ a Sybil attack [4] and quickly generate lots of misleading votes. Figure 1 outlines the network topology for a Sybil attack. Any technique that makes the edge cut between the honest region and Sybil region smaller can be effective in practice A gate-keeper mechanism that disincentivizes joining the network multiple times in rapid succession would alleviate this issue, such as a cryptographic challenge comparable to the proof of work mechanism as employed on certain blockchain structures.

Semantically, Credence assumes a positive vote to be a vote of confidence that the content is authentic; they do not have

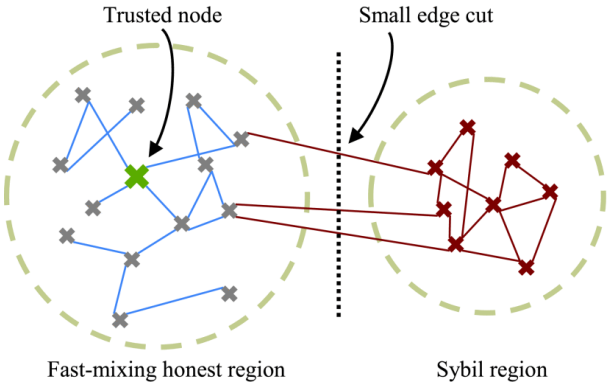

Fig. 1. Sybil region, taken from TODO: Where? Ask Johan

to be indicative of popularity, although in practice users may complect the two. Cooperative peers will generally consistently vote to correctly classify authentic and non-authentic content. Credence uses a weighted average scheme to estimate the authenticity of each content item based on the collected votes. Using the set of content items on which two peers both voted, a local weight is assigned as follows: Assuming that all cooperative peers vote in a similar fashion, looking at the correlation between voting behavior allows a peer to quickly asses the honesty of a different peer.

SpamResist [25] follows a scheme similar to Credence and Sorcery [22], albeit applied to tag clouds. By dividing peers in two groups, the *unfamiliar peers* and the *interacted peers*, two different heuristics with widely different characteristics can be applied. SpamClean [24] is another way to combat tag spam. Again the authors make use of the insight that users with similar voting behavior can provide more reliable information in general, while also leading to degraded performance of the system for deviants and dishonest peers.

SpamLimit [1]...[7].

Scrubber introduces a scheme which allows for swift punishment of malicious users, while still allowing redemption as a result of continuous honest behavior [2]. It operates on the assumption that at least 25% of users react to punishment by removing the polluted content from the network.

The authors of [6] propose a quorum-based approach, where peers can create a quorum to assess the honesty of certain peers based on ad-hoc voting procedure. As long as honest peers participate in these voting schemes this can be effect, but as the authors of [21] noted, user participation in voting is usually nonexistent or wildly inaccurate in the best of cases. This leads us to believe that earlier mentioned schemes, while performing admirably in a simulation or experiment, give no guarantees for real world deployments.

Some special considerations have to be made for preventing dishonest peers from sharing corrupted data with honest peers in live streaming systems; honest peers repeatedly trying to get an uncorrupted copy of the data can shut down the system if left unchallenged. A trivial solution would be to create a published checksum for each unit of data, although this leads to a problematic amount of metadata quickly. The authors of

[5] TODO: Describe [9].

[6] TODO: expand on WoT links and what not.

[7] TODO: Compare SpamLimit vs SpamResist vs SpamClean

[5] introduce a system for honest peers to make a converging assessment regarding the trustworthiness of peer groups by determining and refining probabilities of a specific peer being complicit in repeatedly delivering corrupted data. By taking into account that not each peer is responsible for the same piece of data, this can strike an effective balance between the amount of metadata required and the sensitivity to abuse by dishonest peers.

A. Social enrichment

Most p2p networks have users use information contained within the system to determine which peers to trust. This can be problematic for users who only joined the network recently, as they might not have all the information to correctly categorize peers they interact with. The situation turns into a bootstrap problem, where being initially basing your behavior on information begotten from dishonest peers leads to honest colluding with dishonest peers without by accident.

A different approach allows for using out-of-band information to enrich the information gathered from within the system. One such system is Sorcery as introduced in [22] and [23], adding social network information as a source of baseline truth, thereby addressing the bootstrap problem. Sorcery uses this baseline truth to issue challenges to peers of which no prior knowledge is available. Comparing challenge responses to the baseline truth allows each peer to estimate a relative reliability connected peer in the network.

SybilInfer as proposed in [3] relies on knowledge of all social interaction in a network overlay in order to determine the likelyhood of peers being either honest or dishonest.

IV. Information retrieval in Tribler

Tribler search functionality focuses on three key requirements: fast results, correct results and spam protection [8]. Tribler started out as fork of Yet Another BitTorrent Client, aiming to use social networks to enhance the user experience. Active research has extended Tribler to make it a tool for researchers of decentralized systems to run experiments in the real world. Tribler is compatible with other BitTorrent clients. Users of Tribler can discover content from other peers via a gossip protocol.

A. Communities

Tribler is extendable by introducing communities. A Tribler community is a network overlay via which peers can exchange predefined messages. The SearchCommunity is a Tribler community used to share and receive partial torrent files, allowing users to actually search for content on the Tribler network in a decentralized manner. Each user can have any number of content-channels associated with them. A content channel is essentially a collection of torrent files, as well as an assorted list of subscribers [20]. The AllChannel community stores users' vote preference for content channels, and shares known votes among peers, allowing for filtering based via a distributed moderation system. A vote can either be positive (or "favorite"), or negative (or "spam"), as defined by the VoteCast protocol [9]. Tribler also allows peers to change their mind at a later time and revoke their vote. Changes in voting preferences are propagated over the network via a gossip protocol.

B. Voting data

Tribler stores the current beliefs about of the vote counts per channel in a local database, allowing for offline analysis of voting behaviour within the network. We need an understanding of user voting behavior to evaluate how resilient Tribler is to vote-based spam.

The storage scheme as of this writing allows us to create a coarse overview of how popular each content channel is by calculating an effective vote count per channel. We subtract the number of 'spam' votes from the number of 'favorite' votes for a specific channel, reaching a number of effective votes.

C. Analysis

The VoteCast data crawled from the AllChannel over several hours allows us to see how popularity is distributed over the channels in Tribler. Looking at Figure 2, we see that a channel on average has 127 votes, with only 45 channels having more than 1000 votes. The gathered data leads us to the conclusion that there are a handful of reasonably popular channels, and a myriad of less-popular channels.

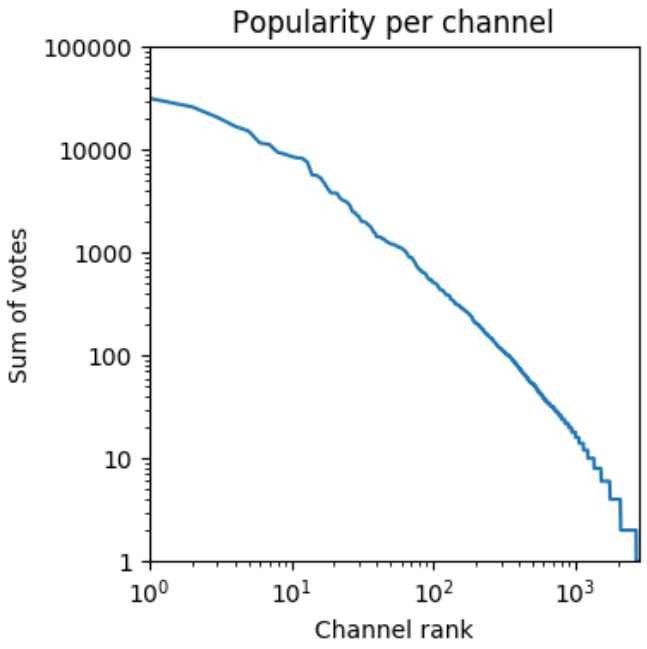

Fig. 2. Content channel popularity, demonstrating that all but the 40 most popular channels only have a handful of subscribers.

[8] https://www.tribler.org/ContentSearch/

[9] https://www.tribler.org/Votecast/

D. Votes over time

Giving a closer look to the gathered VoteCast data reveals that there exist two clusters of votes which predate the existence of Tribler, and by extension VoteCast, by respectively 28 and 6 years. It can be assumed that these anomalies are either the result of either a bug or a curious user testing the limit of the VoteCast validation logic. These findings have no bearing on further elaboration of vote-spam behaviour in Tribler. See Figure 3 for the raw plot of this data, including the anomalous past votes.

Figure 4 shows how votes are distributed when properly filtered. We see that the first few votes were steadily cast, after which the bigger group of users started using the VoteCast system. [10]

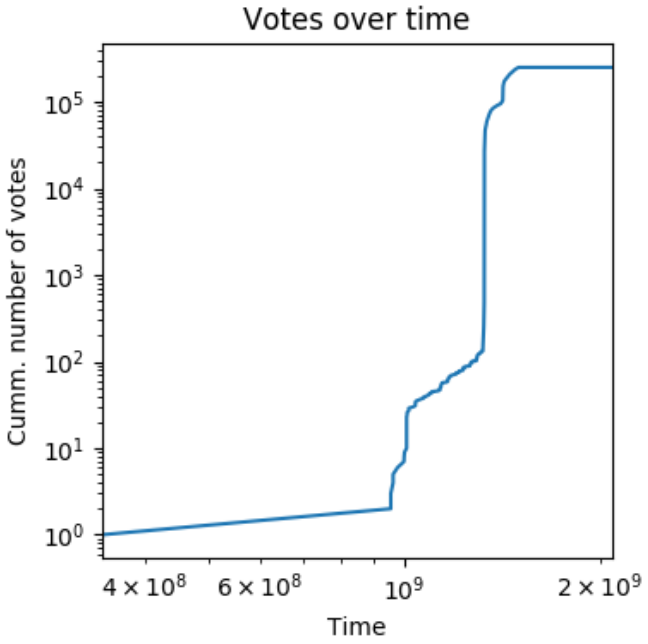

Fig. 3. Voting behaviour over time, unfiltered to include phantom votes.

V. Conclusions

Clearly delineating the set of honest from dishonest peers in a fully decentralized system is a difficult and for the general case unsolved problem. The cold-start problem has newly joined peers having trouble judging the authenticity of claims made by peers in the network, while the lack of an accepted ground truth necessitates heuristics and compromises to establish an initial bearing on trustworthiness. Adding out of band information to assist systems with making these decisions alleviates the cold start problem, but this is not a general solution for systems without persistent identities or systems with high churn.

Honest peers can be misled and contribute to the activities of dishonest peers in various ways. The effects of such poisoned peers in the network should be mitigated, while allowing

[10] http://demo.polr.me/7

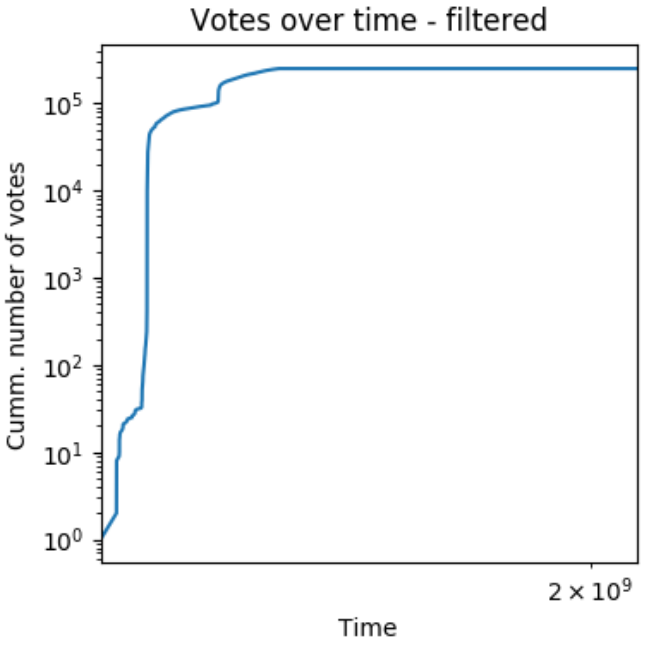

Fig. 4. Voting behaviour over time, filtered to show the distribution of legitimitely cast votes.

poisoned peers redemption in the long term if their behavior is corrected.

The advent of tamper-proof distributed ledger technologies brings with it a renewed interest in creating a Sybil resistant scheme for identifying dishonest peers, by making peers accountable for their behavior.

- [7]
- [13] wrt interaction rules in trust systems
- [17] wrt tag spam

References

[1] Eric Chang. “Defending against Spam in Tagging Systems via Reputations”. In: (2016). URL: http://dlc.dlib.indiana.edu/dlc/handle/10535/10221 (visited on 07/17/2017).

[2] C. Costa and J. Almeida. “Reputation Systems for Fighting Pollution in Peer-to-Peer File Sharing Systems”. In: *Seventh IEEE International Conference on Peer-to-Peer Computing (P2P 2007)*. Sept. 2007, pp. 53–60. DOI: 10.1109/P2P.2007.15.

[3] George Danezis and Prateek Mittal. “SybilInfer: Detecting Sybil Nodes using Social Networks.” In: *NDSS*. San Diego, CA. 2009.

[4] John R Douceur. “The sybil attack”. In: *International Workshop on Peer-to-Peer Systems*. Springer. 2002, pp. 251–260.

[5] Rossano Gaeta, Marco Grangetto, and Lorenzo Bovio. “DIP: Distributed Identification of Polluters in P2P Live Streaming”. In: *ACM Trans. Multimedia Comput. Commun. Appl.* 10.3 (Apr. 2014), 24:1–24:20. ISSN: 1551-6857. DOI: 10.1145/2568223. URL: http://doi.acm.org/10.1145/2568223 (visited on 07/17/2017).

[6] Hatem Ismail, Daniel Germanus, and Neeraj Suri. "P2P routing table poisoning: A quorum-based sanitizing approach". In: *Computers & Security* 65 (2017), pp. 283–299. URL: http://www.sciencedirect.com/science/article/pii/S016740481630178X (visited on 07/17/2017).

[7] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. "The eigentrust algorithm for reputation management in p2p networks". In: *Proceedings of the 12th international conference on World Wide Web*. ACM. 2003, pp. 640–651.

[8] Uichin Lee et al. "Understanding Pollution Dynamics in P2P File Sharing." In: *IPTPS*. Vol. 6. 2006, pp. 1–6. URL: http://netlab.cs.ucla.edu/internal/wiki-internal/files/uclee2006iptps.pdf (visited on 07/16/2017).

[9] Qiao Lian et al. "An empirical study of collusion behavior in the Maze P2P file-sharing system". In: *Distributed Computing Systems, 2007. ICDCS'07. 27th International Conference on*. IEEE, 2007, pp. 56–56. URL: http://ieeexplore.ieee.org/abstract/document/4268209/ (visited on 07/16/2017).

[10] Jian Liang, Naoum Naoumov, and Keith W. Ross. "Efficient blacklisting and pollution-level estimation in p2p file-sharing systems". In: *AINTEC* 3837 (2005), pp. 1–21. URL: http://link.springer.com/content/pdf/10.1007/11599593.pdf#page=10 (visited on 07/17/2017).

[11] Haiman Lin et al. "Conducting routing table poisoning attack in DHT networks". In: *Communications, Circuits and Systems (ICCCAS), 2010 International Conference on*. IEEE, 2010, pp. 254–258. URL: http://ieeexplore.ieee.org/abstract/document/5582015/ (visited on 07/16/2017).

[12] Naoum Naoumov and Keith Ross. "Exploiting p2p systems for ddos attacks". In: *Proceedings of the 1st international conference on Scalable information systems*. ACM. 2006, p. 47.

[13] Johan Pouwelse and Martijn de Vos. "Laws for Creating Trust in the Blockchain Age". unpublished paper. 2017.

[14] Ian Sibner et al. "TorrentTrust: A Trust-Based, Decentralized Object Reputation Network". In: (2016).

[15] Dinh Nguyen Tran et al. "Sybil-Resilient Online Content Voting." In: *NSDI*. Vol. 9. 2009, pp. 15–28. URL: https://www.usenix.org/legacy/events/nsdi09/tech/full_papers/tran/tran_html/ (visited on 07/16/2017).

[16] Kevin Walsh and Emin Gun Sirer. *Thwarting p2p pollution using object reputation*. Tech. rep. Cornell University, 2005. URL: https://ecommons.cornell.edu/handle/1813/5680 (visited on 07/17/2017).

[17] Yongang Wang et al. "Dspam: Defending against spam in tagging systems via users' reliability". In: *Parallel and Distributed Systems (ICPADS), 2010 IEEE 16th International Conference on*. IEEE. 2010, pp. 139–146.

[18] Baoning Wu and Brian D. Davison. "Identifying link farm spam pages". In: *Special interest tracks and posters of the 14th international conference on World Wide Web*. ACM, 2005, pp. 820–829. URL: http://dl.acm.org/citation.cfm?id=1062762 (visited on 07/16/2017).

[19] Quan Yuan et al. "AStudy OF INDEX POISONING IN PEER-TO-PEER FILE SHARING SYSTEMS". In: (). URL: https://pdfs.semanticscholar.org/ef1e/10d6047a1c2f2a07d0214d29092f1270e810.pdf (visited on 07/17/2017).

[20] N. Zeilemaker et al. "Tribler: Search and stream". In: *2011 IEEE International Conference on Peer-to-Peer Computing*. Aug. 2011, pp. 164–165. DOI: 10.1109/P2P.2011.6038729.

[21] Ennan Zhai et al. "Resisting tag spam by leveraging implicit user behaviors". In: *Proceedings of the VLDB Endowment* 10.3 (2016), pp. 241–252. URL: http://dl.acm.org/citation.cfm?id=3021939 (visited on 07/17/2017).

[22] Ennan Zhai et al. "Sorcery: Could we make P2P content sharing systems robust to deceivers?" In: *Peer-to-Peer Computing, 2009. P2P'09. IEEE Ninth International Conference on*. IEEE, 2009, pp. 11–20. URL: http://ieeexplore.ieee.org/abstract/document/5284532/ (visited on 07/16/2017).

[23] Ennan Zhai et al. "Sorcery: Overcoming deceptive votes in P2P content sharing systems". en. In: *Peer-to-Peer Netw. Appl.* 4.2 (June 2011), pp. 178–191. ISSN: 1936-6442, 1936-6450. DOI: 10.1007/s12083-010-0074-2. URL: https://link.springer.com/article/10.1007/s12083-010-0074-2.

[24] Ennan Zhai et al. "Spamclean: Towards spam-free tagging systems". In: *Computational Science and Engineering, 2009. CSE'09. International Conference on*. Vol. 4. IEEE, 2009, pp. 429–435. URL: http://ieeexplore.ieee.org/abstract/document/5284153/ (visited on 07/16/2017).

[25] Ennan Zhai et al. "SpamResist: making peer-to-peer tagging systems robust to spam". In: *Global Telecommunications Conference, 2009. GLOBECOM 2009. IEEE*. IEEE, 2009, pp. 1–6. URL: http://ieeexplore.ieee.org/abstract/document/5425801/ (visited on 07/16/2017).