Foundations of Peer-to-Peer Reputation

Quinten Stokkink
q.a.stokkink@tudelft.nl
Delft University of Technology

Alexander Stannat
a.w.stannat@tudelft.nl
Delft University of Technology

Johan Pouwelse
j.a.pouwelse@tudelft.nl
Delft University of Technology

Abstract

Successful classification of good or bad behavior in the digital domain is limited to central governance, as can be seen with trading platforms, search engines and news feeds. We explore and consolidate existing work on decentralized reputation systems to form a common denominator for what makes a reputation system successful when applied without a centralized reputation authority, formalized in 7 axioms and 3 postulates. Reputation must start from nothing and always reward performed work, respectively lowering and increasing as work is consumed and performed. However, it is impossible for nodes to perform work in a purely synchronous attack-proof work model and real systems must necessarily employ relaxations to such a work model. We show how the relaxations of performing parallel work, allowing unconsumed work and seeding well-known identities with work satisfy our model. Our formalizations allow constraint driven design of decentralized reputation mechanisms.

1 Introduction

The gap between Internet reality and scientific theories is widening. On the one hand decentralized open reputation infrastructure has been investigated by scientists for many decades [16, 13]. The results are numerous proposals, ideas and dozens of surveys. On the other hand, the company eBay has been operating a trustworthy marketplace due to their reputation tracking algorithms for 25 years. We present the first step in closing the gap to allow building Internet platforms which are open, distributed, and fair. Our focus is distributed reputation functions, systematically exploring their restrictions to identify success criteria.

Over the years, big tech has come to dominate our digital lives [27]. But are central parties a hard requirement for digital interactions? Research suggests trust emerges even without a central party. Humans band together based on a common goal and a set of shared norms. This leads to trust and reciprocity, i.e. effort towards this common goal. This is also known as *social capital* [23] and contrary to what the existence of big tech platforms might suggest, this *does* translate to the digital domain [21, 29].

The harsh reality is that, despite the large body of scientific proposals, the notion that big tech monopolies are bad and that the self-organizing effects of social capital emerge, eBay is still here. Scientists have been unable to sufficiently address attacks such as cleaning your identity with whitewashing attacks or creating millions of fake accounts. These are examples of the complexities of decentralization. Decentralized alternatives to big tech have to deal with a plethora of non-trivial attacks [13] and design constraints.

Research that aims to tackle the many problems of decentralization *has* been pragmatic. For example, there even exists a “personalized” variant of Google’s PageRank algorithm that isn’t (necessarily) centralized [11]. However, whereas these solutions have been applied with varying levels of success, they typically do not enjoy long-lived deployment nor do they incrementally improve upon a shared fabric. In order to allow future solutions to construct such a shared fabric, we present restrictions on the design of reputation management solutions. We consolidate existing knowledge in order to move beyond academic ideas, towards proven solutions: we explore the foundations of peer-to-peer reputation.

2 Problem Statement

We suppose that there exists some application that requires nodes to perform certain non-trivial tasks, to benefit this application. In other words, we assume a system where nodes can work together to achieve a common goal (a common good) and honest nodes are externally incentivized to achieve it. Secondly, we assume that nodes, acting within the bounds of our described system, employ a reputation mechanism to evaluate their locally known information to reciprocate. Essentially, this means that nodes are interested in determining their relative *preference* for interactions with their peers in the network. This is closely related to the well-explored *utilty theory* in microeconomics [10], but similar formalizations for decentralized reputation mechanisms are sparse.

In contrast to the large body of existing work surrounding reputation management systems, this paper will not present a new reputation management solution. The scope of this paper will also not be to make a comprehensive list of all attacks on decentralized reputation systems. Instead, this paper identifies 7 axioms and 3 postulates for attack-resilient decentralized reputation mechanisms to be practically viable. In this paper we present as general as possible rules over all reputation mechanisms within a peer-to-peer network setting, bound by our context, definitions and assumptions. Concretely, the contributions of this paper are the following:

- The definitions and motivations of constraints for attack-proof bootstrapping of new identities.
- An impossibility result for the existence of a purely synchronous attack-proof work model.
- The definition of attack-proof relaxations for synchronous work models.
- The definitions and motivations of constraints for building reputation in a decentralized fashion.

3 System Model

We consider a peer-to-peer system consisting of interconnected nodes capable of exchanging messages. We do not assume any node has all information. We do assume that the application establishes a well-formed history of information and all nodes continuously publish new information. This forms an infinite well-formed history: a history with unbounded size and one where subhistories (partial views of the information in the system) do not conflict [5]. A practical example is a blockchain: a linear log of unbounded length. Secondly, we also do not assume all nodes evaluate history in the same way. Each node may employ its own reputation mechanism, which may be further personalized per node.

3.1 Reputation Mechanisms

All reputation mechanisms can essentially be modeled by three components: reporting, work derivation and reputation calculation. The reporting component captures the history of the system, e.g. in blockchains: mining a block makes all of its transactions part of the blockchain's history (a technicality being that you also need to wait 6 more blocks for these transactions to actually become final). Next is assigning value to each of these reports: the derivation of work. The aggregation of net *work* (also referred to as expected *utility*) nodes perform for the network forms a *work accounting* mechanism [26]. Finally, this absolute contribution to the network in homogenized work units can be mapped to a relative reputation score for each locally known node.

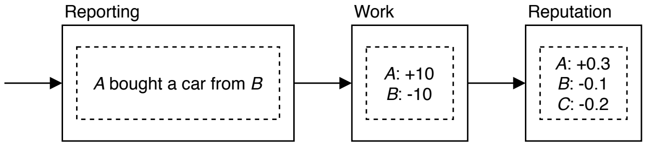

Figure 1: An example of an event propagating through the reporting, work and reputation mechanisms.

An example of events in such a system is given in Figure 1. In this example an event within the scope of the system's semantics (buying cars) is integrated into history by the reporting mechanism, affecting nodes A and B. Both the buying and selling of a car is considered as work. Here work is symmetric: A transfers 10 units of work to B. Finally, the calculated work influences the reputations of the involved nodes and another node C. These scores are not symmetric.

3.2 Reciprocity

Nodes in our model perform tasks for each other, which we call performing work. Initially, we will define this work as the homogenized work unit value of a semantic event in the totally ordered history of the system between two node identifiers. We use $w_{i,j,t}$ to denote the value in work units transferred from a node i to a node j at time t in the history. This definition introduces symmetry in work: $\forall w_{i,j,t} : w_{i,j,t} = -w_{j,i,t}$. In Section 5 we relax the conditions of a totally ordered history and work symmetry of this definition. Our full list of definitions is given in Table 1.

Reputation is used to choose what other node to perform work for, to reciprocate. We assume this choice is *unbiased*, i.e. the calculated reputation does not depend on the node identifier. A node with higher reputation is more likely to receive work from another honest node. Choosing to reciprocate builds reputation and also fosters more trust and reciprocity. Formally, this model has been previously explored and defined as the cycle of reputation, trust and reciprocity [21].

3.3 Attack model

Attacks on reputation systems have been extensively, perhaps comprehensively, reported [17]. Specifically, Sybil attacks are considered the biggest threat and are the primary focus of many reputation mechanism proposals [1]. The goal of all these attacks, within the scope of our model, is the consumption of work without reciprocity. The scope of this paper is not to provide a list of these attacks, nor is the scope of this paper to provide a *complete* blueprint to make reputation mechanisms attacker-proof. Instead, we provide only part of this blueprint, based on the well-known body of identified attacks. We refer the reader to related work for the full

Table 1: Terminology

Symbol	Description
V	The set of node identifiers.
t	An absolute measure of time $\in \mathbb{N}_0$.
$r(i,t)$	A function of node i at time t, giving a reputation value on $[r_a, r_b)$.
$w_{i,j,t}$	The work node i performs for node j at time t.

list of possible attacks, of which we will shortly highlight the recurring ones referenced in this work:

- **Whitewashing attack.** The ability of any user to leave the system and rejoin it with a new identity with the intent of disassociating with its past behavior.
- **Sybil attack.** The ability of any user to create an unbounded amount of identities to subvert the functional requirements of a peer-to-peer solution [9].

4 Bootstrapping of Reputation

In truly decentralized peer-to-peer architectures identity creation is cheap. We do not assume the Sybil attack to be solved and derive the appropriate bootstrapping conditions for newly created identities in a system. Our propositions stem from conjectures in existing works that a starting reputation of 0 for a reputation function on the interval $[0, 1]$ is Sybil-proof [8]. We then show that the result of this is that newly created identities must contribute to the network in order to bootstrap, as supported by existing work [20]. Finally, we show no work is ever performed in such a model.

Axiom 4.1 (Reputation must start from nothing) *Whenever a node i enters the system at a time $t_i \geq 0$ (where 0 is the starting time of the system) it must have the minimum possible reputation value r_a.*

$$\underset{i \in V}{\forall}\ r(i, t_i) = r_a$$

Motivation. Suppose there exists a starting reputation value r_x of a node i, where $r_x > r_a$. Reputation is the normalization and relativization of work given to the network and taken from the network [21]. Consequently, $r_x > r_a$ allows consumption of work from the network [25]. As identity creation is cheap in a decentralized peer-to-peer network, any entity can perform a whitewashing attack to create another identifier j which can also consume work from the network in the same manner [22]. In conclusion, as the starting reputation for any node i is $r_x > r_a$, any entity can consume work proportional to the amount of fake identities the entity creates, without reciprocation. As it must hold that reputation leads to reciprocity [21] and the whitewashing attack allows unreciprocated consumption of work, by contradiction $r_x > r_a$ cannot be true. Therefore, it must hold that $r_x = r_a$ and thus $\underset{i \in V}{\forall}\ r(i, t_i) = r_a$.

As a consequence of Axiom 4.1 all newly created identities should start from the lowest possible value on the reputation function range. If there is either no lowest value or there always exists a lower value than r_a, Axiom 4.1 cannot be satisfied. Therefore, we assert the following postulate.

Postulate 4.1.1 *All reputation functions must be defined on a left-bounded interval.*

Given that new identities must start from the minimum reputation score r_a, consequently (and intuitively) all identities must enter the system without any performed work. We now show the implications of this on work and reciprocity.

Axiom 4.2 (Nodes may not consume more work than they have performed) *Work consumed from the network must always follow work performed for the network.*

$$\underset{w_{j,i,t}}{\forall} \left(w_{j,i,t} \leq \sum_{x \in V, t' \leq t} w_{i,x,t'} \right)$$

Motivation. We follow the construction of the contradiction of Axiom 4.1. Suppose there existed consumed work by a node i such that $w_{j,i,t} > \sum_{x \in V, t' \leq t} w_{i,x,t'}$. This means that node i has consumed more work from the network than it has contributed. If the identity controlling node i performs a whitewashing attack, it has consumed work without having to reciprocate at least an equal amount of work. This violates the essence of reputation mechanisms [21]. Therefore, allowing consumption of more work than is performed must be incorrect. In conclusion, every node must perform work for the network before it consumes it.

From Axiom 4.2 we know that nodes may not consume more work than they have performed. We furthermore know that new identities enter the system without having performed any work. This requires a system where all participants need to perform work and nobody is allowed to consume it. Therefore, we derive the following postulate.

Postulate 4.2.1 *No work is ever performed in a pure decentralized synchronous attack-proof work model.*

5 Relaxations

From Postulate 4.2.1 we know that a pure model of synchronous work leads to a situation where no work is ever performed. As we assumed a system where honest nodes aim to perform work together for a common good, a relaxation of our model is required for work to be performed. We explore three non-mutually-exclusive relaxations of our model to allow performing work without violating Axiom 4.2: allowing more than one interaction at a time, allowing creation of work without a counterparty and allowing well-known identities to start with reputation. We give examples how each of these relaxations manifests in real systems.

5.1 Work Parallelism

Thus far we have assumed that all events leading to work occur at different times t. If we relax this condition and allow multiple events (and therefore work) to occur at the exact same time t, this model is satisfiable again. After all, Axiom 4.2 is satisfied as long as the net work performed for the network is equal to the net work consumed from the network at a particular time t. Practically such a system would be round-based to perform a set of permissible actions at each round t. The downside of this approach is that it requires a consensus protocol to determine which events can be allowed in a round (such that the net work remains 0).

Examples of systems that orchestrate aggregation of multiple events (that do not lead to a net loss of work) are the currency transfer models of cryptocurrencies. Nodes may not transfer more currency than they own and sets of currency transfers that conform to this rule are captured in the system's history periodically using a consensus protocol (in a block with a sequence number in the blockchain) [4].

5.2 Unconsumed Work

So far we have assumed that work is always performed for a counterparty (another node in the network). If we relax this condition and allow work to be performed without a counterparty, our model is satisfiable again. More formally, we introduce special counterparties B_i acting as buffers of work for each node i, without violating the work symmetry condition of $w_{i,j,t} = -w_{j,i,t}$. These buffers are allowed to freely consume work from a node i, but can only perform as much work as they have consumed. This type of model can only be used if work can be done in isolation, in other words a model where resources can be created. In service-type applications (e.g. BitTorrent), where one node can *only* perform work for another node, this relaxation can't be used.

Whereas this relaxation can still satisfy Axiom 4.2, a secondary equation emerges for the set of non-buffer identities $V' = V - \{B_x \mid x \in V\}$, where c is equal to the eventual total amount of unconsumed work in the system:

$$\underset{i \in V'}{\forall} \lim_{T \to \infty} \sum_{t=0}^{T} \sum_{j \in V'} w_{i,j,t} = c$$

When $c \to 0$ performed work is (directly or indirectly) reciprocated (e.g. tit-for-tat [2]). When c is bound by a constant value there is an upper limit to the amount of unreciprocated work in a system (e.g. 21 million bitcoins in Bitcoin[14]), Most systems (e.g. webpage cross-references, ride sharing, auctions, message boards and movie review platforms) employ histories where c can grow to infinity (i.e. an unbounded amount of unconsumed work). Generally, any peer-to-peer system that allows (temporarily) unreciprocated work is open to attacks on the information dissemination layer [25].

5.3 Seeded Work

We have assumed a true peer-to-peer system where *all* identities are cheap. We can relax this condition and allow for some well-known peers, remaining attacker-proof [6]. As long as they cannot be freely created, peers with a-priori reputation can exist. All newly created identities can then still be beholden to Axiom 4.1 and Axiom 4.2 as there is a counterparty in the network to perform work for, so new identities can bootstrap into the system. The downside of this approach is that this system is no longer truly peer-to-peer, but includes nodes that are of a higher status (superpeers).

DNS is an example of a system that employs transitive trust to make nodes trustworthy based on their hierarchy [24]. In DNS any party signing for the integrity of a webpage can be traced back to root authorities, that are universally trusted. Whereas DNS has a binary notion of trust, a more non-binary approach has been applied for reputation mechanisms. A decentralized reputation mechanism that used and popularized *roots of trust* is Advogato [18], inspiring the a priori trust values in EigenTrust for example [15].

6 Building Reputation

In the Section 3 we have shown restrictions on how peer-to-peer systems should bootstrap new identities, with respect to work and reputation. We now show restrictions on how reputation may grow in such a system while staying attack resistant, based on existing work that claims a non-decreasing mapping from work to reputation is desired [12, 3]. Whereas this mapping behaves like a non-decreasing function, we show it is actually not well-defined. Finally, we finish the interval definition of Postulate 4.1.1 (giving the left bound), by providing the right side of reputation functions' interval.

In this section we assume that if the *work parallelism* relaxation was used, the reputation function r is event order independent: for any arbitrary ordering of the work at a certain time t and for all nodes i the calculated reputation at the next time $r(i, t+1)$ remains equal. This implies that for all *net* work at time t, any artificial ordering of *gross* work can be imposed to fit our model's symmetry requirement. For example, given an event e_i and an event e_j at time t, the reputation values $r(i, t+1)$ and $r(j, t+1)$ do not change whether e_i or e_j is included in the history first.

The basis of our following motivations will be Axiom 6.1, stating that it must be possible for the ranking, the relative preference, of nodes must be influencable by performing or consuming work. Essentially, this axiom shows that a fair playing field is required for nodes to engage in truly unbiased peer-to-peer reciprocity. All nodes in the system must be able to relatively gain or lose preference.

Axiom 6.1 (Work is able to influence the ranking of others) *A change in work must exist for a node i that changes the relative ranking of any amount of other nodes.*

Motivation. Given our assumption in Section 3 that reputation mechanisms are unbiased, any reputation value is achievable by any node in the system through performing or consuming work, regardless of how this reputation is computed. This means that for any reputation mechanism that does not map to a trivial group (i.e. the empty set or a single reputation value), a reputation value that is different from other nodes can be achieved by a node through performing or consuming work. As the relative reputation implies a ranking between nodes and it is possible for any node to perform work such that its reputation value exceeds that of others (a trivial example is when others start at a reputation value of 0), any node must be able to perform work such that the relative ranking of all other nodes changes.

Having established a fair playing field for all nodes in the system, we now look at inciting reciprocity. As stipulated in our problem statement, performing work for the network should be encouraged. Honest behavior should make a node a more preferable partner to interact with.

Axiom 6.2 (Performing work must increase reputation) *Any node i performing work for another node j is rewarded with an increase in reputation.*

$$w_{i,j,t+1} > 0 \rightarrow r(i, t+1) > r(i, t)$$

Motivation. Given that performing work must lead to reciprocity and the reputation function r provides a ranking of nodes to reciprocate with, we construct two contradictions.

Firstly, suppose $w_{i,j,t+1} > 0 \rightarrow r(i, t+1) < r(i, t)$. In this case, performing work for the network may lead to a decrease in ranking as compared to another node which has performed less work. As the ranking of the node performing work is never relatively increased, the other node will always be favored. Therefore, work is not reciprocated and $w_{i,j,t+1} > 0 \rightarrow r(i, t+1) < r(i, t)$ cannot be true.

In the second case, suppose $w_{i,j,t+1} > 0 \rightarrow r(i, t+1) = r(i, t)$. In this case, work may be performed without an increase in reputation. As the reputation does not change, the relative ranking between nodes cannot change. This contradicts Axiom 6.1, which states work must be able to influence this ranking. Therefore $w_{i,j,t+1} > 0 \rightarrow r(i, t+1) = r(i, t)$ cannot be true.

Finally, as $w_{i,j,t+1} > 0 \rightarrow r(i, t+1) \geq r(i, t)$ and $w_{i,j,t+1} > 0 \rightarrow r(i, t+1) \neq r(i, t)$, it must hold that $w_{i,j,t+1} > 0 \rightarrow r(i, t+1) > r(i, t)$.

Good behavior must be rewarded, however, conversely, when a node consumes from the system this should diminish its standing. This is not a punishment, but rather a necessary consequence of being able to reward the amount of unreciprocated work associated with a node.

Axiom 6.3 (Consuming work must decrease reputation) *A node i consuming work shouldn't haved increased reputation.*

$$w_{i,j,t+1} < 0 \rightarrow r(i, t+1) < r(i, t)$$

Motivation. Our motivation is analogous to the motivation of Axiom 6.2. Suppose $w_{i,j,t+1} < 0 \rightarrow r(i, t+1) > r(i, t)$. In this case, consuming work will increase reputation and preference for reciprocity. Therefore, the node which performs this work is never reciprocated with, violating our functional requirements. Thus, $w_{i,j,t+1} < 0 \rightarrow r(i, t+1) > r(i, t)$ can't be true and $w_{i,j,t+1} < 0 \rightarrow r(i, t+1) < r(i, t)$ holds.

From Axiom 6.2 and Axiom 6.3 it follows that a strict increase in reputation follows a strict increase in work and a decrease in work causes a decrease in reputation and it seems reputation functions are non-decreasing. However, these functions are actually not well-defined as can be derived from the final case, when no work is performed.

Axiom 6.4 (No work may lead to a loss of reputation) *Any node i not performing work can have a decrease in reputation.*

$$w_{i,j,t+1} = 0 \rightarrow r(i, t+1) \leq r(i, t)$$

Motivation. In the basic case, when no nodes perform work and continue not to perform work, their reputation and relative ranking must remain equal to follow Axiom 4.1. Now suppose $w_{i,j,t+1} = 0 \rightarrow r(i, t+1) \geq r(i, t)$. Consequently, another node, which is not interacted with, may perform work that is not able to lower node i's relative ranking. This contradicts Axiom 6.1, which states it must be able to. Thus, $w_{i,j,t+1} = 0 \rightarrow r(i, t+1) \geq r(i, t)$ can't be true and by extension $w_{i,j,t+1} = 0 \rightarrow r(i, t+1) \leq r(i, t)$ holds.

The fact that reputation functions are not well-defined complicates the design of reputation functions. From these axioms we can, however, derive a postulate which may be useful for reputation mechanism designers:

Postulate 6.4.1 *Any increase in reputation must be earned through performing work.*

$$r(i, t+1) > r(i, t) \rightarrow w_{i,j,t+1} > 0$$

We hereby end our axioms on the growth of reputation regarding the performed and consumed work by nodes. Following the definition of the previous axioms, we finally derive the interval of attack-proof reputation functions.

Axiom 6.5 (Reputation functions must be defined on a right- open left-closed interval) *Any reputation function must be defined on the interval*

$$[r_a, r_b)$$

Motivation. From Postulate 4.1.1 we know reputation functions have a left-bound of r_a. Suppose the range of possible reputation scores were not right-open. Any work performed by a node that has attained the highest possible reputation score, can perform work without gaining reputation. Conversely, this implies this node can then consume the same amount of work without losing reputation. This contradicts Axiom 6.2, as there is no maximum amount of work and as such there can be no highest possible reputation score. In conclusion, the interval must be left-closed and right-open.

By Axiom 6.5, in contrast to related work [8, 19], we find that neither reputation functions defined on $[0, 1]$, nor those defined on $(-1, 1)$ are actually attack-proof.

7 Related Work

Using reputation and trust to incite reciprocity is not a necessity. Events with objective value often allow shortcuts in work accounting. These types of mechanisms are concerned with finding (subgame perfect) Nash equilibria and defining evolutionary stable strategies. An example is the BitTorrent tit-for-tat model [7]. While closely related and complementary to our work, we focus on the reputation mechanism on top of the work derivation and incentive mechanisms.

The normalization in reputation mechanisms is usually not only scaling the work with respect to the sum of other node's work, but usually also involves the social structure in which the work was performed. These mechanisms perform analyses on the social graph of work to detect Sybil regions [28]. Our work is not focused on detecting Sybil regions, but providing rules to avoid vulnerabilities that could be exploited through Sybil attacks.

A framework for testing the Sybil-proofness of reputation mechanisms has been previously created [6]. Concretely, it has been proven that asymmetric reputation functions need (1) diminishing returns, (2) monotonicity and (3) need to be transitive. This partially overlaps with our work, where we refute the frequent claim of monotonicity [12, 3]. While we find reputation functions are similar to non-decreasing functions, ultimately they are not well-defined.

8 Conclusion

This paper has consolidated and has motivated common restrictions for reputation functions. We have provided checks for reputation mechanisms that fit to our synchronous and symmetric work model. New identities in a system must start from no reputation and no work. Reputation functions, mapping work to reputation, must be defined on a left-closed right-open interval and respectively punish or reward work with reputation (though these functions are not well-defined). Hereby, designers have a generalized toolbox of common requirements for the design of reputation mechanisms.

References

[1] Muhammad Al-Qurishi, Mabrook Al-Rakhami, Atif Alamri, Majed Alrubaian, Sk Md Mizanur Rahman, and M Shamim Hossain. 2017. Sybil defense techniques in online social networks: a survey. *IEEE Access* 5 (2017), 1200–1219.

[2] Robert Axelrod and William Donald Hamilton. 1981. The evolution of cooperation. *science* 211, 4489 (1981), 1390–1396.

[3] Moshe Babaioff, John Chuang, and Michal Feldman. 2007. Incentives in peer-to-peer systems. *Algorithmic Game Theory* (2007), 593–611.

[4] Shehar Bano, Alberto Sonnino, Mustafa Al-Bassam, Sarah Azouvi, Patrick McCorry, Sarah Meiklejohn, and George Danezis. 2019. SoK: Consensus in the age of blockchains. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies*. 183–198.

[5] Sebastian Burckhardt. 2014. *Principles ofEventual Consistency*. Vol. 1. Foundations and Trends® in Programming Languages. 1–150 pages.

[6] Alice Cheng and Eric Friedman. 2005. Sybilproof reputation mechanisms. *Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems* (2005), 128–132.

[7] Bram Cohen. 2003. Incentives build robustness in BitTorrent. In *Workshop on Economics of Peer-to-Peer systems*, Vol. 6. 68–72.

[8] Rahim Delaviz, Nazareno Andrade, Johan A Pouwelse, and Dick HJ Epema. 2012. SybilRes: A Sybil-resilient flow-based decentralized reputation mechanism. In *2012 IEEE 32nd International Conference on Distributed Computing Systems*. IEEE, 203–213.

[9] John R Douceur. 2002. The sybil attack. In *International workshop on peer-to-peer systems*. Springer, 251–260.

[10] Peter C Fishburn. 1988. *Nonlinear preference and utility theory*. Number 5. Johns Hopkins University Press Baltimore.

[11] Dániel Fogaras, Balázs Rácz, Károly Csalogány, and Tamás Sarlós. 2005. Towards scaling fully personalized pagerank: Algorithms, lower bounds, and experiments. *Internet Mathematics* 2, 3 (2005), 333–358.

[12] Christopher J Hazard and Munindar P Singh. 2013. Macau: A basis for evaluating reputation systems. In *Twenty-Third IJCAI*. IJCAI.

[13] Kevin Hoffman, David Zage, and Cristina Nita-Rotaru. 2009. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys (CSUR)* 42, 1 (2009), 1–31.

[14] George F Hurlburt and Irena Bojanova. 2014. Bitcoin: benefit or curse? *It Professional* 16, 3 (2014), 10–15.

[15] Sepandar D Kamvar, Mario T Schlosser, and Hector Garcia-Molina. 2003. The eigentrust algorithm for reputation management in p2p networks. In *12th international conference on WWW*. 640–651.

[16] Eleni Koutrouli and Aphrodite Tsalgatidou. 2006. Reputation-based trust systems for P2P applications: design issues and comparison framework. In *Trust, privacy and security in digital business*. Springer.

[17] Eleni Koutrouli and Aphrodite Tsalgatidou. 2012. Taxonomy of attacks and defense mechanisms in P2P reputation systems - Lessons for reputation system designers. *Comp. Sci. Review* 6, 2-3 (2012), 47–70.

[18] Raphael L Levien. 2002. *Attack resistant trust metrics*. Ph.D. Dissertation. University of California at Berkeley.

[19] Michel Meulpolder, Johan A Pouwelse, Dick HJ Epema, and Henk J Sips. 2009. Bartercast: A practical approach to prevent lazy freeriding in p2p networks. In *2009 IEEE International Symposium on Parallel & Distributed Processing*. IEEE, 1–8.

[20] Jacob Jan-David Mol, Johan A Pouwelse, Michel Meulpolder, Dick HJ Epema, and Henk J Sips. 2008. Give-to-get: free-riding resilient video-on-demand in p2p systems. In *Multimedia Computing and Networking 2008*, Vol. 6818. International Society for Optics and Photonics, 681804.

[21] Lik Mui, Mojdeh Mohtashemi, and Ari Halberstadt. 2002. A computational model of trust and reputation. In *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*. IEEE, 2431–2439.

[22] Jordi Nin, Barbara Carminati, Elena Ferrari, and Vicenç Torra. 2009. Computing reputation for collaborative private networks. In *2009 33rd Annual IEEE International COMPSAC*, Vol. 1. IEEE, 246–253.

[23] Robert D Putnam, Robert Leonardi, and Raffaella Y Nanetti. 1994. *Making democracy work: Civic traditions in modern Italy*. Princeton.

[24] Venugopalan Ramasubramanian and Emin Gün Sirer. 2005. Perils of transitive trust in the domain name system. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*. 35–35.

[25] Sven Seuken and David C Parkes. 2014. Sybil-proof accounting mechanisms with transitive trust. *Proceedings of AAMAS* (2014).

[26] Sven Seuken, Jie Tang, and David C Parkes. 2010. Accounting mechanisms for distributed work systems. In *Twenty-Fourth AAAI Conf.*

[27] Aryan Tiwari. 2019. Big Tech Monopoly-Effects, Desirability and Viable Regulations. *CYBERNOMICS* 1, 7 (2019), 19–22.

[28] Bimal Viswanath, Ansley Post, Krishna P Gummadi, and Alan Mislove. 2010. An analysis of social network-based sybil defenses. *ACM SIGCOMM Computer Communication Review* 40, 4 (2010), 363–374.

[29] Yao Wang and Julita Vassileva. 2003. Trust and reputation model in peer-to-peer networks. In *Proceedings Third International Conference on Peer-to-Peer Computing (P2P2003)*. IEEE, 150–157.