

Trustworthy Reputation Games and Applications to Proof-of-Reputation Blockchains

PETROS DRINEAS, Purdue University, pdrineas@purdue.edu

ROHIT NEMA, Stanford University, rohit@rohitnema.me

RAFAIL OSTROVSKY, UCLA, rafail@ucla.edu

VASSILIS ZIKAS, Georgia Institute of Technology, vzikas@gatech.edu

Reputation systems play an essential role in the Internet era, as they enable people to decide whom to trust, by collecting and aggregating data about users' behavior. Recently, several works proposed the use of reputation for the design and scalability improvement of decentralized (blockchain) ledgers; however, such systems are prone to manipulation and to our knowledge no game-theoretic treatment exists that can support their economic robustness.

In this work we put forth a new model for the design of what we call, *trustworthy reputation systems*. Concretely, we describe a class of games, which we term *trustworthy reputation games*, that enable a set of users to report a function of their beliefs about the trustworthiness of each server in a set—i.e., their estimate of the probability that this server will behave according to its specified strategy—in a way that satisfies the following properties:

- (1) It is (ϵ -)best response for any rational user in the game to play a prescribed (truthful) strategy according to their true belief.
- (2) Assuming that the users' beliefs are not too far from the *true* trustworthiness of the servers, playing the above (ϵ -)Nash equilibrium allows anyone who observes the users' strategies to estimate the relative trustworthiness of any two servers.

Our utilities and decoding function build on a connection between the well known PageRank algorithm and the problem of trustworthiness discovery, which can be of independent interest. Finally, we show how the above games are motivated by and can be leveraged in proof-of-reputation (PoR) blockchains.

CONTENTS

Abstract	0
Contents	0
1 Introduction	1
2 Preliminaries	5
3 Trustworthy Reputation Games	8
4 Perfect Information About Nature	12
5 (Consistent) Noisy Information About Nature	14
6 Connection to PoR/PoS Blockchain	17
7 Conclusion and Future Work	18
Acknowledgments	19
References	19

1 Introduction

Trustworthiness plays a central role in security, cryptography, and distributed systems, where the existence of trusted authorities simplifies, if not trivializes, several of the underlying goals. As an example, the primary goal of decentralization is to allow for a secure distributed implementation of a trusted third party. In economics and game theory, reputation has long been used as a tool to allow agents to decide on and reason about the trustworthiness of other agents, and make predictions about their behavior (e.g., in repeated games [Mailath and Samuelson, 2006, Mui, 2002, Resnick et al., 2001]).

However, despite the long-studied relation (in economics) between reputation and trustworthiness, with only a few notable exceptions, reputation has received little attention in the theory of cryptography and distributed computing literature. One reason for this is likely the fact that basing a system’s security on a quantity that is extracted by past observation substantially deviates from the worst-case adversarial model, e.g., makes the system susceptible to attacks from parties who act honestly until they are rendered a key, high-reputation role in the system, and then use this role to break the system’s security.

The recent revolution of blockchain protocols, triggered by the wide adoption of Bitcoin, Ethereum, and other cryptocurrencies, has brought a renewed interest in reputation in the context of such protocols. A reputation system in such protocols is typically used in combination with a more mainstream assumption, like honest majority of hashing power (in *proof of work*, in short, PoW, blockchains) or honest majority of stake (in *proof of stake*, in short, PoS, blockchains). The common methodology here is to interpret reputation of (some of) the nodes as a way to select honest (super)majority committees in a faster and/or more scalable manner. This allows to speed up the block settling time—i.e., the time it takes for a block to be confirmed—and even add *finality*—i.e., valid blocks are confirmed as soon as they are seen on the network—on Nakamoto style blockchains¹, like Bitcoin [Biryukov et al., 2017, Chow, 2007, Gai et al., 2018, Kleinrock et al., 2020, Yu et al., 2019].

Despite several proposed constructions which rely on the existence of *trustworthy reputation systems*, the question of how to allow the blockchain to derive such a reputation system has not been addressed, and existing proposals are restricted to either assuming that such a system is given (e.g., written on the first, so-called *genesis* block of the blockchain [Kleinrock et al., 2020], or devising ad-hoc formulas with little to no justification of why they are right and how they can be computed in a real-world blockchain scenario.

Our work aims to bridge the above gap by proposing an appropriate game theoretic model and a first solution to the above challenging problem. To this direction we use the following methodology: As is common in reputation-based constructions we model trustworthiness as a *reputation system* which is a vector of m independent binary distributions that correspond to the probability of each of the m servers/blockchain-nodes to follow their protocol (i.e., remain “honest”). We refer to the i th value in this vector, R_i , as the *trustworthiness* of the i th server (we also at times refer to the trustworthiness vector as the *ground truth*). Importantly, the ground truth is not encoded on the blockchain. Instead the blockchain users have their beliefs about the trustworthiness of each node. Our goal, then, is to design games that allow the blockchain to extract an order-preserving estimate of this information from its users, assuming they are rational. Our games rely on a novel connection between the behavior of the well-known PageRank algorithm [Brin and Page, 1998] on a bipartite graph and the problem of estimating the trustworthiness of the servers from the beliefs of sufficiently informed rational users, which can be of independent interest.

¹In Nakamoto consensus, a user cannot immediately consider a block as confirmed but needs to wait for it to get deep enough on its local valid chain.

1.1 Our Contribution and Technical Overview

We propose a class of games, which we term *trustworthy reputation games* (in short, *TRep games*) that capture the intuition of the above goal. In a nutshell, a *trustworthy reputation game* is an n -agent Bayesian game against nature, where the (private) state of nature is encoded as a vector of m values in $[0, 1]$, $\mathbf{R} = (R_1, \dots, R_m)$. In our reputation-based blockchain scenario above, m corresponds to the number of blockchain servers/nodes and each R_j correspond to the ground truth (trustworthiness score) of the j th blockchain node. Note that in this work we focus on blockchains with a fixed universe of m nodes. Extending our treatment to dynamically changing universe is an interesting future direction. Notwithstanding, to allow most generality we allow these sets to be disjoint.

As discussed above, our goal is to enable (anyone with read-access to) the blockchain to *discover* \mathbf{R} by observing messages posted by the users in our decentralized scenario. To this direction, the set of pure actions of each user is an endorsement of one of the j nodes; hence a mixed strategy (which is what the user should record on the blockchain to receive associated rewards) is a probability distribution over the set of the m servers. A key feature of trustworthy reputation games is that they come equipped with an efficient *decoding* function, which, given any strategy profile from a given class, computes a specified *reputation* function of the ground truth. Looking ahead, in the games we design, this class will consist of Nash equilibria (NE) and the reputation function will provide an estimate of the *relative* trustworthiness between any two servers, R_i/R_j for $i, j \in [m]$; such an estimate will allow us to order the servers according to their trustworthiness, which can then be used by a reputation-based blockchain in their committee selection.

The core novelty in our work lies in a connection between the well-known PageRank algorithm [Brin and Page, 1998] and our above goals, which we encode into our trustworthy reputation game. The PageRank algorithm was developed to help a search engine to rank web pages so that it can offer better search results, and is considered a catalyst for the dominance of Google among early search engines. It operates by assigning a rank to each page, which is determined by the number and quality of links pointing to it. The core idea is that pages with more incoming links—i.e., pages linked to by more other pages—especially by high-ranking pages, are considered more valuable.

A bit more formally, the idea of PageRank is to model the Internet as a graph, where the nodes are pages and a link in page P pointing to page P' corresponds to a directed edge in this graph from P to P' . The algorithm can be seen as a random walk on this graph, which starts at a random page, and in every step one might either move along one of the outgoing edges of the current node or, with some given probability, “teleport” to some other node. (This last idea was added to ensure that the walk does not get “stuck”.) The (Page)Rank of any node is then computed based on the frequency that the node is visited in comparison to the overall time of the walk. The details of the algorithm are not necessary for understanding the intuition of our game. In our technical section, we offer the details one needs to understand how the parameters of our game are derived from PageRank.

Due to its importance, the PageRank algorithm has been extensively studied and several of its variants have been proposed. Two variants are of particular importance for us: the first one is PageRank over a weighted graph, where for every node u its outgoing edges have non-negative weights that add up to 1 [Xing and Ghorbani, 2004], which can be seen as u ’s weighted *endorsement* of each neighbors; and the other is *Personalized* PageRank [Andersen et al., 2007, Bianchini et al., 2005, Gupta et al., 2013, Iván and Grolmusz, 2010, Yang et al., 2024], which, intuitively, computes the importance of every node u relative to a specific node v , i.e., how relevant the endorsement of u was in v receiving its PageRank.

Here is how we use PageRank to derive the utilities in our game and compute (an order-preserving estimate of) the servers’ trustworthiness scores: consider the unidirectional bipartite graph where

the users are the sources (nodes with incoming degree 0) and the servers are the sinks (nodes with outgoing degree 0). Each user u has an edge pointing to each node/server s_i , $i \in [m]$ whose weight corresponds to the probability according to u that s_i will behave honestly (in other words, it corresponds to u 's belief about the trustworthiness score R_i of s_i .) We refer to this edge as u 's weighted *endorsement* of s_i . Comparing the structure of this bipartite graph to a weighted graph (as used in weighted PageRank), one might observe that an edge from u to s_i (which is the endorser's, u , perception of the endorsee's, s_i , trustworthiness) is analogous to u including a weighted link to s_i , which corresponds to the perception, from u 's point, of how important s_i is in the graph. As such, it is natural to expect that running weighted PageRank on our (bipartite) graph would yield a rank(-score) of each server s_i that is a good estimate of s_i 's trustworthiness from the collective perception of the users (sources in the graph). Hence, if the users have an (approximately) accurate perception of that trustworthiness, PageRank will compute it. We will refer to the rank that is assigned to each server in the above setting as the server's (*computed*) *reputation score*. As we shall see, the above intuition is correct: if the users' beliefs on the servers' trustworthiness are (approximately) accurate, and their endorsements are truthful, then PageRank on the above graph will yield an order-preserving estimate of the ground truth.

The above idea of how to use PageRank on our graph to compute (an estimate of) the ground truth has several issues that one needs to overcome. On the technical side, first, the above bipartite graph is not suitable for PageRank as the random walk will stop in one step, and second, the sum of the outgoing edges of a user is not necessarily 1 (as is required by weighted PageRank). Both these mismatches can be resolved by appropriately "massaging" the graph—adding appropriate teleportation edges and normalizing the weights (we refer to the technical section for details). However, to our knowledge the behavior of the PageRank over such a (massaged) graph has not been sufficiently studied to be able to deduce that the servers' rank is actually their (relative) trustworthiness score. Our analysis demonstrates that this is indeed the case, which we believe can be of independent interest for using this methodology in different scenarios.

This leaves open the following question: even if PageRank works as anticipated when users play according to their true beliefs about the servers' trustworthiness, how can we guarantee they do so? This is where game theory comes to the rescue: we design a Bayesian game, where the (private) state of nature corresponds to the servers' trustworthiness scores (ground truth). The belief of the users about this ground truth is modeled by the user's type. The utility is computed as follows: draw a string from the ground-truth distribution (R_1, \dots, R_n) —as we discuss below, a 1 bit for the j th component corresponds to the j th server following the protocol—and for each server j for which a bit 1 is sampled, we use (an adaptation of) the Personalized PageRank algorithm to reward its endorsers. As we prove the (expected value of the) above utility in our trustworthy reputation game makes playing according to one's true belief an (ϵ -)Nash equilibrium for an ϵ which diminishes in the number of users (in fact, the strategy is a unique Nash if the users' beliefs are perfect).

It is worth mentioning that due to the simplicity of the our bipartite graph, running PageRank and Personalized PageRank on the above (massaged) graph results on relatively simple utility and decoding functions—the decoding function ends up being an average of the weight of incoming edges (endorsements from all users) and the utility of each user is essentially the users' relative contribution to this average. Generalizing the approach to more complex endorsement graphs is in our opinion an excellent future research direction and can expand the methodology to a broader class of applications.

INSTANTIATIONS OF OUR TREP-GAME We instantiate the above methodology in two games that assume different users' beliefs about the servers' trustworthiness, represented as beliefs on the state of nature:

Perfect Information about Nature. We start with the case where users have a perfect information about the state of nature, i.e., every user knows the vector $\vec{R} = (R_1, \dots, R_m)$. This is of course a degenerate form of a TRep Game (it is in fact a complete information game against nature rather than Bayesian). Notwithstanding we find this simpler setting ideal to showcase our methodology, and in particular demonstrate that the above PageRank-based utility and raking system work as anticipated. We prove the following result for this game, $\mathcal{G}_{\text{perfect}}$:

THEOREM 1.1 (INFORMAL). *$\mathcal{G}_{\text{perfect}}$ has a unique Nash equilibrium (NE), where the players play according to their (accurate) beliefs. By observing the players' NE strategies, we can compute a reputation score ρ_i for each server $i \in [m]$, such that for any $i, j \in [m]$, $|\rho_i/\rho_j - R_i/R_j| = \epsilon$.*

Consistent Noisy Information about Nature. We next proceed to a game which is motivated by a more realistic blockchain scenario. We assume that all players have a noisy but *consistent* view of nature's state, i.e., they all know a value $r_j \in [0, 1]$ (for each $j \in [m]$) where with high probability (confidence), r_j is within some ϵ from the trustworthiness score of server j . We consider this as a natural scenario which can, for example, occur by all parties applying a consistent statistic on public data about each server. We prove the following result for this game, $\mathcal{G}_{\text{noisy}}$:

THEOREM 1.2 (INFORMAL). *In $\mathcal{G}_{\text{noisy}}$, playing according to their beliefs/types is ϵ' -Nash for an ϵ' that diminishes with the size of the user set. By observing the players' strategies, we can compute a reputation score ρ_i for each server $i \in [m]$, so that for any $i, j \in [m]$, with high probability, $|\rho_i/\rho_j - R_i/R_j|$ is within an ϵ'' (diminishing in m) factor from R_i/R_j .*

Under assumptions about the density of the trustworthiness score vector and/or the number of servers, the above allows to limit with high probability the number of *inversions* in the ordering derived by the decoded reputation scores.

CONNECTION TO PoR/POS-BLOCKCHAINS. Finally, we discuss how the above games can be used within a reputation-based blockchain, in particular with the hybrid Proof-of-Reputation/Proof-of-Stake blockchain by [Kleinrock et al., 2020] (we provide an overview in Section 2.4).

The idea is that we can associate nature's state (i.e., trustworthiness score of each of each nodes/servers) with the probability that that the node follows the protocol. The blockchain, then, mints for each server who followed the protocol a fixed amount of coins which is distributed according to our (Personalized PageRank-based) utility function.

It is worth mentioning that our treatment applies only to the static reputation setting discussed in [Kleinrock et al., 2020], where the challenge is to discover the initial reputation (e.g., in a bootstrapping phase) and encode it on a PoR genesis block. Extending our treatment to dynamic (updateable) reputation is an interesting research direction.

1.2 Related Work

A number of works have studied reputation as an important concept of behavioral analysis in game theory, (see, e.g., [Mui, 2002, Resnick et al., 2001, Sun, 2015] and references therein.) One of the most important application domains is in repeated and more, generally, sequential games, where reputation typically models information about a player's state that can be extracted from the player's past actions and used to improve the future response to this player [Abreu and Pearce, 2007, Ely and Välimäki, 2003, Fudenberg and Levine, 1992, Mailath and Samuelson, 2006, Schmidt, 1993]. A related line of work investigates learning nature in repeated games [Cripps et al., 2008, Fudenberg and Yamamoto, 2011, Leoni, 2014, Renault and Tomala, 2004, Sugaya and Yamamoto, 2020]. Our work can be seen as combining both threads in a novel way: our goal is to learn the state of nature, which corresponds to an external reputation (as opposed to learning the reputation of the players

themselves) by using the strategies of the player (as opposed to the move of nature). Moreover, we aim to learn this state in a one-shot game which is motivated by a blockchain application.

A number of works have applied game theoretic reasoning to the PageRank algorithm [Foulley et al., 2018, Hopcroft and Sheldon, 2008, Maestre and Ishii, 2016]; however, the goal of these works is rooted in the original use of the algorithm to discover important pages on the Internet and/or important nodes in a communication network. In particular, the goal is for the players (page creators) to maximize some revenue (or a network related quantity like latency of a transmitted message or accuracy of anomaly detection) by strategically pointing to appropriate pages/network-nodes. While we also examine the effect of strategic play on PageRank, we do so in a specially designed game where (Personalized) PageRank is also used in the utilities. Our goal is to instead show that PageRank captures reputation in an extremely natural way (at least in the case of our massaged bipartite graphs) and playing strategically indeed, recovers a meaningful quantity (the relative trustworthiness scores). Discovering other classes of graphs, which enable such reasoning is in our opinion a very interesting direction.

In the context of blockchain, a number of proposals have investigated using an existing reputation system to improve the properties (liveness, finality, and scalability) of consensus and blockchain protocols [Biryukov et al., 2017, Chow, 2007, Gai et al., 2018, Kleinrock et al., 2020, Yu et al., 2019]. These works, however, either do not touch the question of how the blockchain derives such a reputation system, or resort to typically ad-hoc tokenization of reputation, e.g., receiving reputation tokens for hashing or staking. (We refer to [Esber and Kominers, 2021] for a high level discussion on economic considerations of tokenizing reputation). In contrast, in this work we put forth the question, and an appropriate model, for the blockchain extracting information about the trustworthiness of its nodes from its user’s beliefs.

Finally, the structure and goals of trustworthy reputation games bare resemblance to Bayesian optimal design (BOD) [Nisan et al., 2007]. However there are key differences: in contrast to BOD, (1) we (the designer) do not know the distribution of the agent’s valuation (types), and (2) our objective is to learn the ground truth (nature’s state), rather than these types or their distribution. Similarly, to our knowledge, techniques from *Prior-independent mechanisms* (PIMs) [Azar et al., 2019, Devanur et al., 2011, Hartline and Roughgarden, 2009], which are designed for the incomplete information setting, do not apply here as the objectives are different and the game is a one-shot game.

2 Preliminaries

In this section we introduce basic notation used throughout our technical sections, and provide the relevant background on the PageRank algorithm and (reputation-based) blockchains.

2.1 Notation

Let \mathbb{Z} denote the set of integers, and $\mathbb{Z}_{\geq k}$, the set of integers greater than or equal to k . For $n \in \mathbb{Z}_{\geq 1}$, let $[n]$ equal the set, $\{1, 2, \dots, n\}$.

For any probabilistic event E , let the indicator of E , $\mathbf{1}[E]$ denote the binary random variable that outputs 1 when E occurs and 0 otherwise.

We write vectors in boldface, as in \mathbf{v} , $\boldsymbol{\pi}$, \dots ; and matrices in capital letters. Denote the $n \times m$ matrix of all ones as, $\mathbf{1}_{n \times m}$. Then, $\mathbf{1}_{n \times 1}$ is the row-vector of all one’s of length n . Often when referring to the coordinates of some vector, \mathbf{v} , we drop the boldface. Therefore, the i th coordinate of \mathbf{v} is v_i . We index matrices using $M[i, j]$ to denote the value in i th row and j th column.

Let \mathbf{e}_i denote the vector whose i th coordinate, denoted by $e_{i,i}$ is 1, and 0 otherwise.

In this work, we will often talk about L_1 -normalized vectors, by which we mean vectors that have been scaled such that their L_1 -norm—sum of all entries—sum up to 1. Specifically, let $\|\cdot\|_1$ be

the L_1 norm. We define an L_1 normalizing function, $N: [0, 1]^n \rightarrow [0, 1]^n$ such that,

$$N(\boldsymbol{v}) = \left(\frac{v_1}{\|\boldsymbol{v}\|_1}, \dots, \frac{v_m}{\|\boldsymbol{v}\|_1} \right)$$

We call $N(\boldsymbol{v})$ the L_1 -normalized vector of \boldsymbol{v} and denote its j th coordinate by $N(\boldsymbol{v})_j$.

2.2 PageRank

We present the *weighted* graph version of PageRank [Brin and Page, 1998, Xing and Ghorbani, 2004] which readily generalizes the unweighted link graph structure in which it was initially proposed. Intuitively, one can view weighted PageRank (on a weighted graph) as standard PageRank where the weights are captured by adding more links (proportionally to the weights). In applications, such weights correspond to some real number indicating “trust”, “importance” or quality of the outgoing link. We assume weights are non-negative.

Let $G = (V, E)$ be a weighted, directed graph. Let $|V| = n$. Write $V = (v_1, \dots, v_n)$ with respect to an appropriate indexing. We have $(u, v, w) \in E$ if there exists an edge from u to v with (non-negative) weight, w . Let $w_{\text{out}}(v)$ be the sum of the weight of all outgoing edges of $v \in V$. We assume $w_{\text{out}}(v) > 0$ for all $v \in V$ since PageRank is not well-defined for “dangling nodes”. Let $w_{\text{out}}(V)$ equal the vector of $w_{\text{out}}(v_i)$ for all $v_i \in V$, and W_{out} be the $n \times n$ diagonal matrix with diagonal equal to $w_{\text{out}}(V)$. Let M be the adjacency matrix of G , with respect to the same indexing of V .

Fix *restart probability* constant, $\alpha \in (0, 1)$. The PageRank vector, $\boldsymbol{\pi}$ is the solution to the following equation,

$$\boldsymbol{\pi} = \boldsymbol{\pi} (1 - \alpha) W_{\text{out}}^{-1} M + \frac{\alpha}{n} \cdot \mathbf{1}$$

where $\mathbf{1}$ is the vector of all 1’s and with the constraint that $\|\boldsymbol{\pi}\|_1 = 1$. We overload notation and denote $\boldsymbol{\pi}(v_i) = \pi_i$ as the *PageRank*, aka the *rank*, of v_i .

We can also model PageRank as the stationary distribution of a row-stochastic markov chain,

$$M' = (1 - \alpha) W_{\text{out}}^{-1} M + \frac{\alpha}{n} \cdot \mathbf{1}_{n \times n}$$

i.e., the PageRank vector is the solution to,

$$\boldsymbol{\pi} = \boldsymbol{\pi} M'$$

Observe that $W_{\text{out}}^{-1} M$ is the L_1 -row-normalized adjacency matrix of G and can be viewed as the transition matrix of G where the probability of transition is directly proportional to the relative weight on the outgoing edge. The row-stochastic matrix $\frac{1}{n} \mathbf{1}_{n \times n}$ captures the idea of *restarting* uniformly to any vertex in the graph. The convex combination of row-stochastic matrices is also row-stochastic and therefore, M' is row-stochastic.

When are we guaranteed that the stationary distribution (the PageRank vector) $\boldsymbol{\pi}$ exists? We present the following lemma whose proof can be found in most standard elementary textbook on stochastic processes:

LEMMA 2.1 ([HÄGGSTRÖM, 2002, LEVIN ET AL., 2006]). *A finite markov chain has a unique stationary distribution if its transition matrix, M is irreducible. If the chain is also aperiodic then the limiting distribution converges to the stationary distribution.*

It is not hard to see that M' is irreducible and aperiodic for any G (due to *restart*) and therefore, has a unique stationary distribution invariant of what distribution you start with. Therefore, the PageRank vector is well-defined.

Later, it will be helpful to examine PageRank as a system of linear equations. In particular, we can rewrite the above equation as,

$$\pi(v_i) = \sum_{j \in [n]} \pi(v_j) \cdot M'[i, j]$$

2.3 Personalized PageRank and Contribution

PageRank is an emergent global property of the graph. After taking in all edges into consideration, it calculates a global “ranking” or importance score for each vertex. *Personalized PageRank* [Brin and Page, 1998] aims to capture the (bidirectional) relationship between any two vertices in the graph; it can be seen as the “significance” of some target node, t with respect to some source node, s or also, the “importance” of s from the perspective of t [Andersen et al., 2007, Bianchini et al., 2005, Gupta et al., 2013, Iván and Grolmusz, 2010, Yang et al., 2024]. In this work, we will be more interested in the latter interpretation as we will measure the importance of users in the eventual ranking of servers and distribute rewards proportional to the relative importance (which we later call, *contribution*) of each user.

Formally, the Personalized PageRank vector for vertex s , $\pi_s = (\pi_{s,1}, \dots, \pi_{s,n})$ assigns a numeric score to each v_i in the graph. Let $s = v_k$. We overload notation again and denote, $\pi_s(v_i) = \pi_{s,i}$. For the same restart-probability constant as in PageRank, π_s is the (unique) solution to the following equation,

$$\pi_s = \pi_s(1 - \alpha)W_{\text{out}}^{-1}M + \alpha e_k$$

with $\|\pi_s\|_1 = 1$.

Personalized PageRank can be interpreted as a random walk over the graph that starts at s and with probability $1 - \alpha$ travels to a neighboring vertex along the path of the random walk, and with probability α restarts at s . Then, the significance of t with respect to s is how frequently the random walk passes through the vertex, t and is equal to, $\pi_s(t)$. In other words, we can also model Personalized PageRank as the stationary distribution of a row-stochastic markov chain,

$$M'_s = (1 - \alpha)W_{\text{out}}^{-1}M + \alpha E_s$$

where E_s is the $n \times n$ matrix in which every row is equal to e_s .

As mentioned, Personalized PageRank can also be interpreted from the view of t . Define the *inverse* Personalized PageRank vector, aka *Contribution* PageRank vector, for vertex t , π_t^{-1} as the vector of $\pi_s(t)$ for all $s \in V$. In other words, $\pi_t^{-1}(s) = \pi_s(t)$.

We define, the *relative contribution* PageRank vector for t as,

$$\omega_t^{-1}(s) = \frac{\pi_t^{-1}(s)}{\sum_{s \in V} \pi_t^{-1}(s)}$$

Later, we will be interested in measuring the relative contribution with respect to only a subset of the vertices, V' which we define by only summing over the contributions from V' ,

$$\omega_{t|V'}^{-1}(s) = \frac{\pi_t^{-1}(s)}{\sum_{s \in V'} \pi_t^{-1}(s)}$$

Personalized PageRank can also be generalized to any distribution of source nodes by changing e_k in the equation. For example, we can measure the significance of t with respect to a set of vertices, S , of size m instead of a single vertex by replacing e_s with a vector that is equal to $1/m$ at each coordinate that corresponds to S . Intuitively, this corresponds to the random walk restarting from any vertex in S uniformly at random. Then, Personalized PageRank of t with respect to S is how frequently a random walk starting (and restarting) from S passes through t .

2.4 Proof of Reputation with Proof of Stake Fallback

In Section 6, we will show how our trustworthy reputation games can be applied to the *proof-of-reputation with proof-of-stake fallback* (in short, PoR/PoS) paradigm for bootstrapping—i.e., setting the initial parameters of—the blockchain proposed in [Kleinrock et al., 2020]. Below, we provide a high level overview of that paradigm.

The PoR/PoS paradigm assumes a so-called *reputation system*, which is a vector of n independent (probability distributions on) binary random variables (R_1, \dots, R_n) . The number n is the number of blockchain nodes (also referred to as *servers*) who are tasked with running the blockchain protocol, e.g., in Bitcoin that would be the *miners*. Each R_i corresponds to the probability that the i th node will remain “honest” in the protocol, i.e., it will follow its prescribed protocol. Given such a reputation vector, as long as it closely captures the “ground truth,” or in other words, the true probability of servers’ honesty, [Kleinrock et al., 2020] shows how to select committees of size $C = \text{polylog}(n)$ so that with big probability a majority of the parties in all committees will be honest.

As argued in [Asharov et al., 2013, Kleinrock et al., 2020] the simplest way for achieving the above goal (maximizes the probability of honest majorities) is to order the parties according to their reputation, and select the top C in this ordering. (In fact, the actual mechanism/lottery for this selection has several additional properties, which ensures and intuitive notion of fairness; but the above simple deterministic selection algorithm is sufficient for understanding the use of reputation within PoR blockchains.)

Given such a mechanism for selecting honest-majority committees, a PoR blockchain can be constructed in a similar way as common Byzantine Fault-Tolerant (BFT) blockchains, e.g., Algorand in the proof-of-stake (PoS) setting: proceed in phases (often referred to a *slots* or *blockchain rounds*) where in each i th round a committee is chosen to vote on (by adding their digital signature on) the i th block, and a block is accepted if and only if its voted by more than $C/2$ parties in this i th-slot committee (i.e., has a majority vote by committee members).

Importantly, observe that in order to take a decision on whether or not the majority of the i th slot committee has voted, it is essential that parties can verify whether a signature corresponds to a party in this committee. And to make sure that parties adopt the same block, there should be agreement among the committee members. This is done by making sure that the reputation system (and the associated randomness) used for the lottery are “known” to the blockchain (i.e., they are encoded in its past blocks).

The blockchain from [Kleinrock et al., 2020] also fortifies the security of its above PoR methodology by assuming a fallback blockchain, which is based on the proof-of-stake paradigm, and is used to detect and correct forks due to an inaccurate reputation system. In particular, parties running the PoR based construction above, periodically report (a publicly verifiable digest) of their view. The proposed mechanisms ensures that if the blockchain properties are violated then it will be promptly noted on the secondary chain. In this case, the system (temporarily) falls back to that secondary chain. As discussed above, this mechanism is necessary for security; however, its details beyond what is discussed above are not relevant for our paper.

3 Trustworthy Reputation Games

In this section, we formally define a new class of games called *Trustworthy Reputation Games*. We show how to use PageRank to define meaningful utilities, and a “meta-objective” of the game, which we formalize using the notion of *decodability*, which we introduce to our games.

As a reminder, the goal of the model is: users have some belief about the trustworthiness of the servers and we wish for them to act in accordance to their beliefs. As a game, we model the trustworthiness scores of the servers as *nature’s* (private) state and the belief as a *type*. We thus,

model our game as a Bayesian game [Harsanyi, 1982, Zamir, 2009] where nature assigns a type to each player representing each players' belief about nature's private state before the game begins. We formalize this below.

Definition 3.1 (Trustworthy Reputation Game with (\mathcal{E}, f) -Decodability). Let $n, m \in \mathbb{Z}_{\geq 2}$.

The Game. A *Trustworthy Reputation Game* or *TRep Game* is a *simultaneous* Bayesian game against nature and is defined as a tuple, $\mathcal{G} = \left(\mathcal{P}, \mathcal{A} = \prod_{i \in [n]} \mathcal{A}_i, (u_i)_{i \in [n]}, (T_i)_{i \in [n]}, (R_j)_{j \in [m]} \right)$ where,

- (i) **Players.** $\mathcal{P} = (P_i)_{i \in [n]}$ is the set of n agents/players,
- (ii) **Action Space.** \mathcal{A} is the action space (or equivalently pure strategy space since the game is simultaneous) of all the players. Every player has the same set of actions, $\mathcal{A}_i = [m]$, and so $\mathcal{A} = [m]^n$. Let $a_i = j$ if P_i picks j as its action. Let Δ_i be the set of all probability distributions over \mathcal{A}_i , i.e., the set of all (mixed) strategies of P_i . Strategies are represented as an m -vector of probabilities. Let $\Delta = \prod_{i \in [n]} \Delta_i$, the set of all strategy profiles of the game.
- (iii) **Nature.** $(R_j)_{j \in [m]}$ is nature's (private) state. Each $R_j \in [0, 1]$ is interpreted as a probability. We treat Nature as a non-strategic player that always plays the same strategy and has no payoff from the game. Nature's move is determined by flipping m biased coins or Bernoulli random variables, $H_j \sim \text{Ber}(R_j)$, i.e., $\Pr[H_j = 1] = R_j$. Denote by $\mathcal{N} = \{0, 1\}^m$, the space of nature's move, and $\Delta_{\mathcal{N}}$, the set of all probability distributions over \mathcal{N} . So, nature will always play the (mixed) strategy, $\mathbf{R} = (R_1, \dots, R_m) \in \Delta_{\mathcal{N}}$.
- (iv) **Types.** Players may not have perfect information about nature's state which we capture by assigning a "type" to each player. Let T_i be the type space of P_i . $t_i \in T_i$ will represent P_i 's belief about nature's state. (In the following sections, we will give instantiations of the general game by considering different type spaces.) Denote by $\mathcal{T} = \prod_{i \in [n]} T_i$, the space of all players' types.
- (v) **Utilities.** $u_i: \mathcal{T} \times \mathcal{A} \times \mathcal{N} \rightarrow [0, 1]$ is the utility of P_i and depends on the players' types, actions and nature's move. When referring to the (expected) utility with respect to mixed strategies, we overload notation as is standard and write, $u_i: \mathcal{T} \times \Delta \times \Delta_{\mathcal{N}} \rightarrow [0, 1]$. We formally define the utilities in the next section.
- (vi) **Decodability.** Let $\mathcal{E} \subseteq \Delta$ be some set of possible strategy profiles of \mathcal{G} , and let $f: [0, 1]^m \rightarrow [0, 1]^*$ be a possibly randomized function with codomain as vectors of arbitrary length over $[0, 1]$. We say that a TRep game, \mathcal{G} is (\mathcal{E}, f) -decodable if there exists an *efficient* decoding function, $\mathcal{D}: \Delta \rightarrow [0, 1]^*$ such that when sampled using any $\mathbf{e} \in \mathcal{E}$, is identical to f sampled using nature's private state, i.e.,

$$\mathcal{D}(\mathbf{e}) \simeq f(R_1, \dots, R_m)$$

We call f the *reputation* function, and we say, the set of strategy profiles, \mathcal{E} , f -encodes nature's private state.

In words, the idea of the game is for the players' to encode some function over nature's private state with a suitable set of strategy profiles. They may have some belief about nature's state *a priori* which influences their utility and strategies.

Before we introduce the utilities, we translate the structure of our game into a graph which formalizes the blockchain network model we are interested in.

Definition 3.2 (Trustworthy Reputation Graphs). Let $G = \left(V = \mathcal{V} \sqcup \hat{\mathcal{V}}, E, \mathbf{R} \right)$ be a bipartite, directed, edge-weighted, partially vertex-weighted graph, where \mathbf{R} is the vector of vertex weights

on $\hat{\mathcal{V}}$ only, and $(u, v, w) \in E$ iff there exists an edge from $u \in \mathcal{V}$ to $v \in \hat{\mathcal{V}}$ with weight w . We denote by $\mathcal{V} = \{u_1, \dots, u_n\}$ the set of n users and by $\hat{\mathcal{V}} = \{v_1, \dots, v_m\}$ the set of m servers.

G is a *Trustworthy Reputation Graph* or *TRep Graph* if,

- (i) edges are directed from \mathcal{V} to $\hat{\mathcal{V}}$ and therefore, vertices in $\hat{\mathcal{V}}$ are *sinks* (i.e., out-degree is 0),
- (ii) weights are non-negative and for all $u \in \mathcal{V}$, $\sum_{(u, \cdot, w)} w = 1$, i.e., the weights on the outgoing edges from each vertex, respectively, sum up to 1,
- (iii) $\mathbf{R} = (R_1, \dots, R_m)$, and $R_j \in [0, 1]$ for all $j \in [m]$, i.e., the weight on each vertex in $\hat{\mathcal{V}}$ is in $[0, 1]$.

For each $u_i \in \mathcal{V}$, let $\mathbf{w}^{(i)}$ be the vector of edge weights to $\hat{\mathcal{V}}$. So $w_j^{(i)}$ is equal to w if $(u_i, v_j, w) \in E$ and 0 otherwise. We call this vector, $\mathbf{w}^{(i)}$ the *endorsements* of u_i .

For each $v_j \in \hat{\mathcal{V}}$, call R_j the *trustworthiness score* of v_j .

In effect, the edge weights from each vertex in \mathcal{V} are modeled as a probability distribution over the vertices in $\hat{\mathcal{V}}$. It is easy to see the correspondence between TRep games and TRep graphs. The mixed strategies of each player correspond to the outgoing weighted edges from \mathcal{V} (with respect to some indexing), and nature's state corresponds to the vertex weights.

We are now ready to introduce the PageRank-inspired utility and decoding function that underlies TRep games.

3.1 Defining Utilities and Decoding Using PageRank

We examine the (weighted) PageRank algorithm on TRep Graphs. First, we show how PageRank can be used to evaluate servers. Next, we show a corresponding utility function, defined using Personalized PageRank, that when used in TRep games enable PageRank's evaluation to recover the (relative) trustworthiness of the servers.

3.1.1 Evaluating Servers. Let G be a TRep graph. We cannot use such a graph structure readily with PageRank due to the issue of *dangling* vertices. Observe that every vertex in $\hat{\mathcal{V}}$ is dangling (out-degree 0) due the directed unidirectional nature of the graph. We resolve this issue by adding a self-loop to each vertex in $\hat{\mathcal{V}}$.

We make a modification to the general PageRank algorithm reminiscent of Personalized PageRank. In the case where the random surfer decides to restart, instead of teleporting to any vertex uniformly at random, we restrict its teleportation to \mathcal{V} uniformly at random. That is, a random walk will always restart from \mathcal{V} . The intuition behind this modification is analogous to (generalized) Personalized PageRank—in our model, we are only interested in the importance of the servers, $\hat{\mathcal{V}}$, derived from the users; more specifically, we are interested in the relative contribution PageRank of the users for each server.

Let T be the transition matrix for G with the above self-loop alteration. We arrange the vertices such that the first n indices correspond to \mathcal{V} and the next m correspond to $\hat{\mathcal{V}}$. Then, we wish to calculate,

$$\boldsymbol{\pi} = \boldsymbol{\pi} ((1 - \alpha) T) + \alpha \left[\frac{1}{n} \cdot \mathbf{1}_{n \times 1} \quad \mathbf{0}_{m \times 1} \right]$$

Let T' be the Markov chain underlying the above relation (i.e., including restart). We remark that $\boldsymbol{\pi}$ exists and is unique. Observe that clearly every user is *accessible* from any server. As for the other way around, there are two cases, (1) for all $v \in \hat{\mathcal{V}}$, there exists $u \in \mathcal{V}$ such that there is an edge from u to v . In that case, the underlying Markov chain is clearly irreducible; (2) there exists $v \in \hat{\mathcal{V}}$, such that for all $u \in \mathcal{V}$, there is no edge from u to v . In that case, v is a *transient* state and its long term distribution approaches 0. In some sense, v is irrelevant to the long term behavior of the Markov chain and can be ignored. So long as the entire graph is not transient states, we can prune

all such transient servers where their probability is 0 in the stationary distribution; the remainder of the states will constitute an irreducible Markov chain (with aperiodicity due to self-loops) for which there exists a unique stationary distribution. Note that this fits our intuition precisely— v 's *importance* as indicated by the stationary distribution is 0 as no user *trusts* it.

For the prescribed graph structure, we can calculate the PageRank of each server explicitly using the system of linear equations. Let $\Pr[u_i \rightarrow v_j | T] = w_j^{(i)}$, the probability of transitioning from vertex u_i to v_j as specified in T . Note that $\Pr[v \rightarrow v' | T] = 0$, and $\Pr[v \rightarrow v | T] = 1$ for any $v \neq v'$, $v, v' \in \hat{\mathcal{V}}$. Also, for $u \in \mathcal{V}$, $\Pr[u \rightarrow v | T'] = (1 - \alpha) \Pr[u \rightarrow v | T]$ as with probability α we restart to \mathcal{V} . Therefore, for each $v \in \hat{\mathcal{V}}$, we have,

$$\begin{aligned} \pi(v) &= \Pr[v \rightarrow v | T'] \pi(v) + \sum_{u \in \mathcal{V}} \Pr[u \rightarrow v | T'] \cdot \pi(u) \\ &= (1 - \alpha) \Pr[v \rightarrow v | T] \pi(v) + (1 - \alpha) \sum_{u \in \mathcal{V}} \Pr[u \rightarrow v | T] \cdot \pi(u) \\ &= (1 - \alpha) \pi(v) + (1 - \alpha) \sum_{u \in \mathcal{V}} \Pr[u \rightarrow v | T] \cdot \pi(u) \\ \implies \pi(v) &= \frac{1 - \alpha}{\alpha} \sum_{u \in \mathcal{V}} \Pr[u \rightarrow v | T] \cdot \pi(u) \end{aligned}$$

By symmetry in T' , $\pi(u) = \pi(u') = c$ for all $u, u' \in \mathcal{V}$ and thus,

$$\pi(v) = c \frac{1 - \alpha}{\alpha} \sum_{u \in \mathcal{V}} \Pr[u \rightarrow v | T]$$

Therefore, the PageRank of each server is directly proportional to the sum of its incoming *endorsements* from the users, \mathcal{V} . Since we are only interested in the PageRank of the servers, we normalize on the set of the servers which is equivalent to taking the average of the incoming endorsements, and define, the “reputation score” of server j , $v_j \in \hat{\mathcal{V}}$ as,

$$\rho_j = \frac{1}{n} \sum_{u \in \mathcal{V}} \Pr[u \rightarrow v | T] \tag{1}$$

Normalizing yields the convenient property that the reputation scores sum up to 1, since,

$$\sum_{v_j \in \hat{\mathcal{V}}} \rho_j = \sum_{v \in \hat{\mathcal{V}}} \left(\frac{1}{n} \sum_{u \in \mathcal{V}} \Pr[u \rightarrow v | T] \right) = \frac{1}{n} \sum_{u \in \mathcal{V}} \sum_{v \in \hat{\mathcal{V}}} \Pr[u \rightarrow v | T] = \frac{1}{n} \sum_{u \in \mathcal{V}} 1 = \frac{1}{n} \cdot n = 1$$

Later, we show how assuming rational players, TRep games—with the utilities as below—ensure the reputation scores correspond to the relative trustworthiness.

3.1.2 Utilities as a Function of Contribution. Similarly, we analyze *Contribution PageRank* on T and observe that, $\pi_v^{-1}(u)$ for $u \in \mathcal{V}, v \in \hat{\mathcal{V}}$ is directly proportional to $\Pr[u \rightarrow v | T]$. Due to the self-loop, v also has a contribution toward itself, and this indeed appears as, $\pi_v^{-1}(v) = 1$; however we are only interested in the (relative) contribution of users and therefore, we use the relative contribution with respect to only \mathcal{V} ,

$$\omega_{v|\mathcal{V}}^{-1}(u) = \frac{\Pr[u \rightarrow v | T]}{\sum_{u' \in \mathcal{V}} \Pr[u' \rightarrow v | T]} \tag{2}$$

This quantity is what we will use when determining the utilities of each player. Recall that in our model, the servers perform some prescribed task such that the behavior/correctness of the server can be evaluated. Specifically, we restrict ourselves to Bernoulli random variables with probabilities

from \mathbf{R} . For each server that behaves “correctly”, we distribute a fraction of unit *reward* proportional to the “contribution” of the user to the server’s reputation score.

And so, we derive the (expected) payoff of user u in the graph as,

$$\sum_{v \in \mathcal{V}} R_j \cdot \omega_{v|\mathcal{V}}^{-1}(u) = \sum_{v \in \mathcal{V}} R_j \frac{\Pr[u \rightarrow v | T]}{\sum_{u' \in \mathcal{V}} \Pr[u' \rightarrow v | T]}$$

When translating this to TRep games, the transition probabilities map to (mixed) strategies. The trustworthiness of the servers map to nature’s private state which fixes nature’s strategy as \mathbf{R} .

While players have beliefs about the trustworthiness of the servers (i.e., nature’s state in the game), their utility does not depend on it. In full generality, the types can be used to establish a distribution over nature’s state enabling Bayesian probability to be used in the analysis of the best response. Thus, we omit the types from the domain for brevity.

Let $\mathbf{s} = (\mathbf{s}^{(1)}, \dots, \mathbf{s}^{(n)}) \in \Delta$ be a strategy profile for the players, and $\mathbf{r} = (r_1, \dots, r_m) \in \mathcal{N}$ be nature’s strategy (which will be fixed). Then, we define the (expected) utility over mixed strategies for $P_i, u_i: \Delta \times \Delta_{\mathcal{N}} \rightarrow [0, 1]$,

$$\mathbb{E}[u_i(\mathbf{s}, \mathbf{r})] = \sum_{j \in [m]} r_j \cdot \frac{s_j^{(i)}}{\sum_{k \in [n]} s_j^{(k)}}$$

Denote $s_j^{(-i)} := \sum_{k \in [n]; k \neq i} s_j^{(k)}$, the sum of every other players’ probability for action j . Hence,

$$\mathbb{E}[u_i(\mathbf{s}, \mathbf{r})] = \sum_{j \in [m]} r_j \cdot \frac{s_j^{(i)}}{s_j^{(i)} + s_j^{(-i)}} \quad (3)$$

3.2 TRep Games Under Different Classes of Beliefs

We study the general TRep games by specifying different distributions of types (classes of beliefs) held by the players. For each distribution, we present an appropriate \mathcal{E} and f for which we can prove decodability. We reason that these choices are natural and highlight a powerful use case in PoR blockchains. Looking ahead, we will use the same decoding function for both games. Precisely, the decoding function will be the reputation score of the servers we derived using PageRank.

Looking ahead, in our analysis, the different distributions of types we study are,

- (1) There is only one type: (R_1, \dots, R_m) and is identically assigned to every player. This represents the case where every player has perfect information about nature (Section 4),
- (2) There is an infinite number of types of the form, $(R_1 \pm \epsilon, \dots, R_m \pm \epsilon)$ for some small $\epsilon > 0$ but every player still has the same type. This is analogous to modeling the players beliefs as an *additive noisy signal* over nature’s state (Section 5).

4 Perfect Information About Nature

In this section, we study the first of the two type distributions mentioned above—every player has the same type that accurately capture nature’s private state. Equivalently, every player has perfect information about nature’s private state, the probability of its coin-flips, R_j . In this case, we define the type space for every P_i as $T_i = T_{\text{perfect}} = [0, 1]^m$. We denote this TRep game as $\mathcal{G}_{\text{perfect}}$. Observe that $\mathcal{G}_{\text{perfect}}$ can be thought of as a (non-bayesian) game against nature with complete and perfect information, where every player aims to maximize, $\mathbb{E}[u_i(\cdot, \mathbf{R})]$. (The expectation is over nature’s randomness and the players’ strategies.)

We make a few observations,

Observation 4.1. $\mathcal{G}_{\text{perfect}}$ is (totally) symmetric, and thus a symmetric Nash equilibrium (NE) exists [Nash, 1951].

Observation 4.2. $\mathcal{G}_{\text{perfect}}$ is (expected) constant-sum, with $\sum_{j \in [m]} R_j$ when players act rationally. What we mean here is that the total utility available to all players (the pot) is fixed in expectation. While there exists strategies where some of amount of the pot is “wasted”—e.g. if $R_1 > 0$ but no player picks 1 as their action— this is not rational assuming mixed strategies as a player (P_i) can assign a tiny probability to action $a_i = 1$ and strictly improve their expected utility (even if it’s split with other players). In other words, if any player plays a strategy such that the probability for each action j (where $R_j \neq 0$) is positive, then the game is constant-sum.

We show that $\mathcal{G}_{\text{perfect}}$ is $(\mathcal{E}_{\text{NE}}, f_1)$ -decodable for,

- \mathcal{E}_{NE} equal to the set of all (expected) Nash equilibria of the game, and
- f_1 equals N , the L1-normalizing function.

Observe that f_1 preserves ratios. That is, $f_1(\mathbf{R}) = N(\mathbf{R})$, and

$$\frac{N(\mathbf{R})_i}{N(\mathbf{R})_j} = \frac{R_i}{R_j}$$

In words, any (expected) NE of the TRep game, $\mathcal{G}_{\text{perfect}}$ encodes the pairwise ratios of the components of nature’s private state.

We first prove the following lemma about the space of equilibria of this simplified game,

LEMMA 4.3. $\mathcal{G}_{\text{perfect}}$ has a unique (expected) NE. The equilibrium is the symmetric strategy profile, $\mathbf{s}^* = (\mathbf{s}^{*(1)}, \dots, \mathbf{s}^{*(n)})$, where $\mathbf{s}^{*(i)} = N(\mathbf{R})$ for all i . Therefore, $\mathcal{E}_{\text{NE}} = \{\mathbf{s}^*\}$.

PROOF. Let $L = (\sum_{j \in [m]} R_j) / n$. Since the game is symmetric, L is an upper bound on the (expected) payoff achievable by any player in a Nash equilibria. First, we show that every player can guarantee a minimum (expected) payoff of L .

Fix P_i and its strategy, $\mathbf{s}^{(i)} = N(\mathbf{R})$.

As a function of the strategies of the other players, the expected utility,

$$\mathbb{E} \left[u_i \left(N(\mathbf{R}), \mathbf{s}^{(-i)}; \mathbf{R} \right) \right] = \sum_{j \in [m]} \frac{N(\mathbf{R})_j \cdot R_j}{\mathbf{s}_j^{(-i)} + N(\mathbf{R})_j} \quad (\text{using (3)})$$

We observe that the Hermitian of the above function is negative definite and therefore is strictly convex as a function of $\mathbf{s}_j^{(-i)}$ ’s over the domain,

$\left\{ \left(\mathbf{s}_1^{(-i)}, \dots, \mathbf{s}_m^{(-i)} \right) : \forall j \in [m], \mathbf{s}_j^{(-i)} \geq 0, \text{ and } \sum_{j \in [m]} \mathbf{s}_j^{(-i)} = (n-1) \right\}$, which is bijective to $\Delta^{(-i)} = \prod_{j \in [m], j \neq i} \Delta_j$, the strategy space of all other players.

Therefore, $\mathbb{E} [u_i]$ has a unique global minimum with respect to the $\mathbf{s}_j^{(-i)}$ ’s. Partially differentiating in each variable, we find that the minimum is achieved at, $\mathbf{s}_j^{(-i)} = (n-1)N(\mathbf{R})_j$. And therefore the minimum value of $\mathbb{E} [u_i(\mathbf{R}; N(\mathbf{R}), \mathbf{s}^{(-i)})]$ is,

$$\sum_{j \in [m]} \frac{N(\mathbf{R})_j \cdot R_j}{\mathbf{s}_j^{(-i)} + N(\mathbf{R})_j} = \sum_{j \in [m]} \frac{N(\mathbf{R})_j \cdot R_j}{(n-1)N(\mathbf{R})_j + N(\mathbf{R})_j} = \sum_{j \in [m]} \frac{R_j}{n} = L$$

Therefore, an (expected) payoff of L is achievable by every player by playing $N(\mathbf{R})$ as its strategy. (It is *non-exploitable*.) Moreover if the other players play any strategy such that there exists j and $\mathbf{s}_j^{(-i)} \neq (n-1)N(\mathbf{R})_j$, then playing $N(\mathbf{R})$ guarantees payoff strictly greater than L . As any Nash cannot exceed individual payoff of more than L , we must have that if $\mathbf{c} = (\mathbf{c}^{(1)}, \dots, \mathbf{c}^{(n)})$ is Nash then, $\mathbf{c}_j^{(-i)} = (n-1)N(\mathbf{R})_j$ for all $i \in [n], j \in [m]$.

This determines a full-rank linear system whose solution is $c_j^{(i)} = \mathbf{N}(\mathbf{R})_j$, which is exactly \mathbf{s}^* . This implies, that if a Nash exists, it must be \mathbf{s}^* .

We can conclude that \mathbf{s}^* must be Nash since we know that there exists a (symmetric) NE in symmetric games as per the observation above. This concludes the proof of the lemma.

Recall that strategies in TRep games map to endorsements in TRep graphs. For each server j , we computed a reputation score, ρ_j that was directly proportional to its PageRank, which simplified to taking an average over incoming endorsements. Translating back to TRep games, we define the *PageRank (PR) decoding* function, $\mathcal{D}_{\text{PR}}: \Delta \rightarrow [0, 1]^m$,

$$\mathcal{D}_{\text{PR}}(\mathbf{s}) = \frac{1}{n} \cdot \left(\sum_{i \in [n]} s_1^{(i)}, \dots, \sum_{i \in [n]} s_m^{(i)} \right)$$

i.e., \mathcal{D}_{PR} takes an average of the (mixed) strategies and is the TRep-game analogue of computing the reputation score.

Finally, we show,

THEOREM 4.4. $\mathcal{G}_{\text{perfect}}$ is $(\mathcal{E}_{\text{NE}}, f_1)$ -decodable using \mathcal{D}_{PR} .

PROOF. Using Lemma 4.3, we have $\mathcal{E}_{\text{NE}} = \{(\mathbf{N}(\mathbf{R}), \dots, \mathbf{N}(\mathbf{R}))\}$, which implies,

$$\begin{aligned} \mathcal{D}_{\text{PR}}(\mathcal{E}_{\text{NE}}) &= \left\{ \frac{1}{n} \cdot \left(\sum_{i \in [n]} \mathbf{N}(\mathbf{R})_1, \dots, \sum_{i \in [n]} \mathbf{N}(\mathbf{R})_m \right) \right\} \\ &= \{(\mathbf{N}(\mathbf{R})_1, \dots, \mathbf{N}(\mathbf{R})_m)\} \\ &= \{\mathbf{N}(\mathbf{R})\} \\ &= f_1(\mathbf{R}) \end{aligned}$$

Therefore, equilibria in TRep games can be decoded using PageRank to the relative values of nature's state (equivalently, to the relative trustworthiness of the servers).

We remark that $\mathcal{G}_{\text{perfect}}$ has the neat property that every players' strategy is exactly the function on nature's state we would like to decode. Therefore, the decoding function could have also been to simply use any players' strategy in a Nash equilibrium as the output. While this is indeed true, this is merely a coincidence due to the simplified nature of the above game, which makes the symmetric NE being unique. The PR decoder provides a unifying treatment in which we use PageRank for both the utilities and decoding, and we believe this methodology can extend to more complicated graph structures.

5 (Consistent) Noisy Information About Nature

We continue with our study of TRep games and in this section explore decodability when every player has an approximation of nature's private state, $\mathbf{R}' = (R'_1, \dots, R'_m)$. We suppose that $R'_j \in [0, 1]$ is distributed such that its expectation as a random variable is equal to R_j , and there exists a constant, $\epsilon > 0$, and some probability, p , such that $\Pr [R'_j \notin [R_j - \epsilon, R_j + \epsilon]] \leq p$. We model it as such to capture the idea of confidence intervals when approximating a random variable. E.g., the sample mean is centered around the true value and is normally distributed. Write, $R'_j = R_j + \delta_j \epsilon'$ for $\delta_j \in \{-1, 1\}$.

We define the type space for every player, P_i , as $T_i = T_{\text{approx}} = [0, 1]^m$. We assume that every player has the *same* type, $\mathbf{t}_i = \mathbf{t}_j = \mathbf{R}' = (R'_1, \dots, R'_m)$, i.e. every player has the same belief about nature's state. Denote this game as $\mathcal{G}_{\text{noisy}}$.

As we shall see, ϵ and p will determine the “slack” in the relative trustworthiness that we're able to decode.

In Section 4 above, we showed that in the perfect information setting ($\mathcal{G}_{\text{perfect}}$), where every player a priori precisely knew nature's private state, the unique NE encoded the relative values of nature's private state, \mathbf{R} . This is of course still true in $\mathcal{G}_{\text{noisy}}$ since the utilities have not changed; however, the players operate with incomplete information and so, we must deal with a weaker notion than Nash.

Instead, we show that TRep games with the noisy but identical type distribution shown above are $(\mathcal{E}_{\text{tt}}, f_2)$ -decodable for,

- \mathcal{E}_{tt} is the singular set of the “truth-telling” strategy profile, denoted by \mathbf{s}_{tt} , in which every players play $N(\mathbf{t}_i)$ as their strategy. We call the strategy $N(\mathbf{t}_i)$ as “truth-telling” as it represents some fixed function of the players' belief about the state of nature, which it can locally compute.
- $f_2: [0, 1]^m \rightarrow [0, 1]^m$ is a randomized function that takes as input $\mathbf{u} = (u_1, \dots, u_m)$, and outputs \mathbf{v} such that $\Pr \left[\mathbb{E} [\|\mathbf{v} - N(\mathbf{u})\|_\infty] \leq \frac{\epsilon}{\|\mathbf{R}'\|_1} \right] \geq 1 - mp - q$, where q is an upper bound on the probability that $|\|\mathbf{R}'\|_1 - \|\mathbf{R}\|_1| \geq \delta$ for some small constant, δ . In the end of the section, we will show that q is exponentially-decreasing in m . In words, with high probability, f_2 , in expectation, outputs a vector “close” to the L_1 -normalized vector with respect to the L_∞ -norm.

In the following we show that \mathbf{s}_{tt} , is an ϵ' -(Nash) equilibrium [Nisan et al., 2007] for $\mathcal{G}_{\text{noisy}}$, where ϵ' is decreasing in n/m^2 .

LEMMA 5.1. *Assuming $\epsilon = O(1/n)$, the “truth-telling” strategy profile, \mathbf{s}_{tt} , is ϵ' -NE for $\epsilon' = O(m^2/n)$.*

PROOF. All players playing $N(\mathbf{R})$ guarantees a payoff that is at most ϵ' less than the payoff from the best response. P_i 's (expected) utility as a function of everyone else playing truthfully is,

$$\mathbb{E} [u_i(\mathbf{x}, \mathbf{s}_{\text{tt}-i}; \mathbf{R})] = \sum_{j \in [m]} \frac{x_j \cdot R_j}{(n-1)N(\mathbf{R}')_j + x_j}$$

The above function is strictly concave on the domain of \mathbf{x} : $\{x_i \geq 0; \sum_{i \in [n]} x_i = 1\}$, and therefore obtains its maximum at a unique point. Partially differentiating and simplifying yields maximum value at \mathbf{x}^* ,

$$x_j^* = n \frac{\sqrt{R_j R'_j}}{\sum_{k \in [m]} \sqrt{R_k R'_k}} - (n-1)N(\mathbf{R}')_j$$

and thus, the deviation from the truthful strategy in each component,

$$\Delta x_j = x_j^* - N(\mathbf{R}')_j = n \frac{\sqrt{R_j R'_j}}{\sum_{k \in [m]} \sqrt{R_k R'_k}} - n \frac{R'_j}{\|\mathbf{R}'\|_1}$$

We coarsely approximate ϵ' . For $\epsilon = O(1/n)$, Δx_j is small, and we can approximate the change in utility, $\epsilon' = \mathbb{E} [u_i(\mathbf{x}^*, \mathbf{s}_{\text{tt}-i}; \mathbf{R})] - \mathbb{E} [u_i(\mathbf{s}_{\text{tt}}; \mathbf{R})]$ using linear approximation. We obtain,

$$\epsilon' \leq \frac{m^2(n-1)}{n^2} \frac{1+\epsilon}{1-\epsilon}. \text{ This completes the proof of the lemma.}$$

We remark that the above is an asymptotic bound that aims at showing that for small enough ϵ , ϵ' is inversely proportional to n/m^2 . In particular, for our blockchain-motivated scenario, where a

natural assumption is that $n \gg m$ (as is the case in major cryptocurrencies) ϵ' will be monotonically decreasing as the system scales. We note in passing that our numerical experiments demonstrate that the value of ϵ' would typically be much smaller than m^2/n , even for a large number of users. This is consistent with our intuition that in our game (where utilities are normalized) playing one's beliefs guarantees a payoff close to the best possible payoff one can hope even when everyone is perfectly informed; therefore small deviations from this strategy should not yield major gains in utility. We conjecture that playing the truth-telling strategy is an *approximate Bayesian Nash Equilibrium* for a tighter ϵ' (dominated by ϵ and p). We leave this as a question for future work.

Now, unlike $\mathcal{G}_{\text{perfect}}$, we cannot perfectly decode $f_1(\mathbf{R}) = \mathbf{N}(\mathbf{R})$ since the players only know an approximation of nature's state. Instead, we prove decodability for the "weaker" function, f_2 , that only approximates the L_1 -normalized function in the L_∞ -norm. Again, we will use the same decoding function, D_{PR} to show that PageRank approximately decodes to the relative trustworthiness scores of the servers.

We first show that $\Pr[|\|\mathbf{R}'\|_1 - \|\mathbf{R}\|_1| \geq \delta] \leq q$ for some q that is exponentially small in m using Hoeffding's bound. Formally, we treat R'_j as a random variable. By assumption, we have $\mathbb{E}[R'_j] = R_j$. Also, $\|\mathbf{R}'\|_1 = \sum_{j \in [m]} R'_j$ by definition, which implies

$$\mathbb{E}[\|\mathbf{R}'\|_1] = \mathbb{E}\left[\sum R'_j\right] = \sum \mathbb{E}[R'_j] = \sum R_j = \|\mathbf{R}\|_1. \quad (4)$$

Therefore, $\Pr[|\|\mathbf{R}'\|_1 - \|\mathbf{R}\|_1| \geq \delta] = \Pr[|\|\mathbf{R}'\|_1 - \mathbb{E}[\|\mathbf{R}'\|_1]| \geq \delta]$, which by Hoeffding's bound [Hoeffding, 1963],

$$\Pr[|\|\mathbf{R}'\|_1 - \mathbb{E}[\|\mathbf{R}'\|_1]| \geq \delta] \leq \exp\left(-\frac{\delta^2}{4\epsilon^2 m}\right) = q$$

which is exponentially decreasing in m . Since the above holds for any δ , for sufficiently large m , we can approximate,

$$\|\mathbf{R}'\|_1 \approx \mathbb{E}[\|\mathbf{R}'\|_1] \stackrel{\text{Eq. (4)}}{=} \|\mathbf{R}\|_1. \quad (5)$$

Using the above, we show,

THEOREM 5.2. $\mathcal{G}_{\text{noisy}}$ is $(\mathcal{E}_{\text{tt}}, f_2)$ -decodable.

PROOF. We have, $\mathcal{E}_{\text{tt}} = \{(\mathbf{N}(\mathbf{R}'), \dots, \mathbf{N}(\mathbf{R}'))\}$, which implies, $\mathcal{D}_{\text{PR}}(\mathcal{E}) = \{\mathbf{N}(\mathbf{R}')\}$

We calculate,

$$\mathbb{E}[\|\mathcal{D}_{\text{PR}}(\mathcal{E}_{\text{tt}}) - \mathbf{N}(\mathbf{R})\|_\infty] = \mathbb{E}[\|\mathbf{N}(\mathbf{R}') - \mathbf{N}(\mathbf{R})\|_\infty] = \max_{j \in [m]} \mathbb{E}[|\mathbf{N}(\mathbf{R}')_j - \mathbf{N}(\mathbf{R})_j|]$$

We restrict ourselves to the intersection of events, $E = \{R'_j \in [R_j - \epsilon, R_j + \epsilon] \text{ for all } j \in [m]\}$, and $H = \{|\|\mathbf{R}'\|_1 - \mathbb{E}[\|\mathbf{R}'\|_1]| \leq \delta\}$.

Now, $\Pr[E \cap H] \geq 1 - \Pr[E^c \cup H^c] \geq 1 - (\Pr[E^c] + \Pr[H^c])$, where A^c is defined the complement of event A . By union bound, $\Pr[E^c] \leq mp$, and, $\Pr[H^c] \leq q$ as we showed above. Thus, $\Pr[E \cap H] \geq 1 - mp - q$.

Conditioned on $E \cap H$, we have $\epsilon' \leq \epsilon$ and,

$$\begin{aligned} \max_{j \in [m]} \mathbb{E}[|\mathbf{N}(\mathbf{R}')_j - \mathbf{N}(\mathbf{R})_j|] &\leq \max_{j \in [m]} \mathbb{E}\left[\left|\frac{R_j + \delta_j \epsilon'}{\|\mathbf{R}'\|_1} - \frac{R_j}{\|\mathbf{R}\|_1}\right|\right] \\ &\stackrel{\text{Eq. (5)}}{\approx} \max_{j \in [m]} \mathbb{E}\left[\left|\frac{R_j + \delta_j \epsilon'}{\|\mathbf{R}\|_1} - \frac{R_j}{\|\mathbf{R}\|_1}\right|\right] \\ &\leq \frac{\epsilon}{\mathbb{E}[\|\mathbf{R}\|_1]} \end{aligned}$$

We conclude that $\mathbb{E} [\|\mathcal{D}_{\text{PR}}(\mathcal{E}_{\text{tt}}) - \mathbf{N}(\mathbf{R})\|_{\infty}] \leq \frac{\epsilon}{\mathbb{E} [\|\mathbf{R}\|_1]}$ with probability at least $1 - mp - q$.

Intuitively, the above result demonstrates that for sufficiently small p , i.e., sufficiently high confidence on the interval, as the system scales in m , the decoding of the above approximate Nash becomes order preserving with respect to the ground truth. In fact, assuming sufficient large gaps in the ground truth, the above will be true even for small values of m .

6 Connection to PoR/PoS Blockchain

We show how our TRep games can be applied to PoR blockchains, e.g., [Kleinrock et al., 2020]. Since the novelty of our work is not the application but rather TRep games themselves, the purpose of this section is to demonstrate the connection, rather than providing the concrete blockchain construction. For this reason, we will keep the discussion informal.

Recall that in a (byzantine-fault tolerant) PoR blockchain like [Kleinrock et al., 2020], a committee of nodes is selected for proposing and voting on each block. Most blockchain-based cryptocurrency systems, give rewards (in terms of coins) to the members of such committee to incentivize participation. However, the rest of the users of the system typically, do not receive any rewards. An exception here is systems that offer their users “dividends” or interest for participation in order to boost adoption and/or availability (e.g., Algorand) and Proof-of-Stake blockchains that support stake delegation (e.g., Cardano, Ethereum, etc.) where users can delegate their stake to stake-pools and they get a fraction of their rewards when their stake pool (operator) is selected to propose the next block. We conjecture that our proposed mechanisms can be used also to incentivize truthful stake delegation. We view this as an interesting research question, albeit less relevant than reputation in a PoR system, as untruthfully extracted reputation can hurt the systems security, whereas untruthful stake delegation does not: if the majority of stake is in honest hands, then honest parties will delegate truthfully, otherwise, security cannot be ensured.

Here, we show how to encode our payoff function into the PoR blockchain for rewarding the users/endorsers in ways that allow everyone to sufficiently estimate their belief of the ground truth. Consistently with [Kleinrock et al., 2020] we will assume a static trustworthy reputation system—i.e., the trustworthiness of the nodes is fixed (part of the ground truth) from the protocol’s onset. At the beginning of the system’s execution, a biased coin is (privately) flipped for each node s_i which is heads with probability R_i . If the outcome of the coin is heads, then s_i is counted as corrupted/Byzantine. We will refer to parties who are not corrupted as *honest*. We remark that an honest party follows the protocol, whereas a byzantine party might not. For the purpose of this analysis, we will assume that all byzantine-corrupted parties are *eventually faulty*, in the sense that there is a bound $\lambda \in \mathbb{N}$ such that all corrupted parties will be faulty during λ rounds of the protocol. We find the assumption of eventual-faultiness natural, as corrupted parties who remain honest have no negative effect on the security properties of the blockchain.

We note that static reputation and party set implies that discovery of the ground truth corresponds to the so-called *bootstrapping* of the blockchain, in particular, creation of its parameters that can be engraved on its genesis block, which that is agreed upon by all parties (users and nodes). Indeed, the reputation system in [Kleinrock et al., 2020] is actually part of their PoR blockchain’s genesis block written on both the PoR and the (fallback) PoS blockchain. Therefore, in the following we focus on how to perform such bootstrapping.

An important consideration here is that in absence of publicly agreed trustworthy reputation, agreeing on such genesis-block parameters is impossible without further assumptions. This is where the fallback PoS change becomes handy in the bootstrapping phase, as the bootstrapping happens on the PoS blockchain.

In the following we show how to use our game(s) to perform this bootstrapping. For simplicity we focus this discussion on the perfect information game $\mathcal{G}_{\text{perfect}}$. Recall that $\mathcal{G}_{\text{perfect}}$ assumes that users have perfect knowledge of the state of nature. At the onset of the protocol, all users play $\mathcal{G}_{\text{perfect}}$ and record their strategies on the PoS blockchain. The bootstrapping protocol then proceeds as follows: we execute the PoR/PoS protocol, with the difference that instead of using the lottery from [Kleinrock et al., 2020] for picking committees, we follow a fixed round-robin schedule: order all nodes lexicographically (e.g., according to their wallet address); in the first slot choose the first $c = \text{polylog}(m)$ servers, then the next c and so on.

It is important to note that in this phase, we have no information about reputation so the safety and/or liveness of the PoR chain might be violated in several slots. The fallback property of the PoR/PoS chain from [Kleinrock et al., 2020], guarantees that when this happens, it is detected on the secondary chain and at least some corrupted node is discovered. When this happens, we restart the bootstrapping with this node excluded. The above process is done until some iteration completes λ (the eventual-faultiness parameter) rounds.

At the end of this phase, the system looks at all information on the bootstrapping phase recorded on the PoS blockchain and rewards users according to $\mathcal{G}_{\text{perfect}}$, where servers detected as malicious correspond to nature sampling 0 and the remainder as sampling 1. Finally, the system executes the decoding function D_{PR} and the associated reputation scores are adopted as the nodes' reputation. This completes the bootstrapping and the blockchain can then start running the protocol from [Kleinrock et al., 2020] with the extracted reputation scores as its reputation system.

The following is a corollary of Lemmas 4.3 and 5.1 by observing that the assumption is that a node s_i is honest with probability R_i and every corrupted node will be faulty (and recorded as corrupted) in the bootstrapping phase—this follows directly by the eventual-faultiness assumption. The above means that the rewards for parties will occur with the same probability as when nature moves in $\mathcal{G}_{\text{perfect}}$

COROLLARY 6.1. *Assuming the users have perfect (resp. consistent noisy) information on the nodes' trustworthiness, playing according to their truth-telling strategy is a Nash equilibrium (resp. ϵ' -best response for ϵ' as in Lemma 5.1).*

The decodability property of the games then ensures that the blockchain can compute an ordering of the servers according to their trustworthiness, which has no inversions in the perfect information case and a small (negligible in m when p is negligible in m) probability of inversions in the incomplete information case. Using this ordering, if selecting the top $\ell = \text{polylog}(m)$ nodes yields a committee where the number of corrupted parties is at most $(1/2 - c'')\ell$, for some constant c'' (which is the assumption in [Kleinrock et al., 2020]) then selecting the top 0.9ℓ parties will also yield an honest majority committee with overwhelming probability. We defer to [Kleinrock et al., 2020] for more details.

7 Conclusion and Future Work

We introduced a class of games, called trustworthy reputation games, which is motivated by the problem of a blockchain system publicly extracting its users' collective perception of the trustworthiness of the blockchain nodes. Our games make a novel use of the PageRank algorithm on bipartite graphs for extracting reputation, which we believe can be of independent interest. Our work opens a number of interested questions, including how to extend our treatment to repeated games, as a means of allowing dynamically updateable reputation systems, and investigating the behavior of our PageRank model on different classes of graphs.

Acknowledgments

Petros Drineas, Rafail Ostrovsky, and Vassilis Zikas were supported in part by the Möbby.ai project.

References

- Dilip Abreu and David Pearce. 2007. Bargaining, Reputation, and Equilibrium Selection in Repeated Games with Contracts. *Econometrica* 75, 3 (May 2007), 653–710. <https://doi.org/10.1111/j.1468-0262.2007.00765.x>
- Reid Andersen, Christian Borgs, Jennifer Chayes, John Hopcraft, Vahab S. Mirrokni, and Shang-Hua Teng. 2007. Local Computation of PageRank Contributions. In *Algorithms and Models for the Web-Graph*, Anthony Bonato and Fan R. K. Chung (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 150–165.
- Gilad Asharov, Yehuda Lindell, and Hila Zarosim. 2013. Fair and Efficient Secure Multiparty Computation with Reputation Systems. In *Advances in Cryptology - ASIACRYPT 2013 - 19th International Conference on the Theory and Application of Cryptology and Information Security, Bengaluru, India, December 1-5, 2013, Proceedings, Part II (Lecture Notes in Computer Science, Vol. 8270)*, Kazuo Sako and Palash Sarkar (Eds.). Springer, 201–220. https://doi.org/10.1007/978-3-642-42045-0_11
- Pablo D. Azar, Robert Kleinberg, and S. Matthew Weinberg. 2019. Prior independent mechanisms via prophet inequalities with limited information. *Games and Economic Behavior* 118 (2019), 511–532. <https://doi.org/10.1016/j.geb.2018.05.006>
- Monica Bianchini, Marco Gori, and Franco Scarselli. 2005. Inside PageRank. *ACM Trans. Internet Technol.* 5, 1 (Feb. 2005), 92–128. <https://doi.org/10.1145/1052934.1052938>
- Alex Biryukov, Daniel Feher, and Dmitry Khovratovich. 2017. Guru: Universal Reputation Module for Distributed Consensus Protocols. Cryptology ePrint Archive, Paper 2017/671. <https://eprint.iacr.org/2017/671>
- Sergey Brin and Lawrence Page. 1998. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Comput. Networks* 30, 1-7 (1998), 107–117. [https://doi.org/10.1016/S0169-7552\(98\)00110-X](https://doi.org/10.1016/S0169-7552(98)00110-X)
- Sherman S. M. Chow. 2007. Running on Karma - P2P Reputation and Currency Systems. In *Cryptology and Network Security, 6th International Conference, CANS 2007, Singapore, December 8-10, 2007, Proceedings (Lecture Notes in Computer Science, Vol. 4856)*, Feng Bao, San Ling, Tatsuaki Okamoto, Huaxiong Wang, and Chaoping Xing (Eds.). Springer, 146–158. https://doi.org/10.1007/978-3-540-76969-9_10
- Martin W. Cripps, Jeffrey C. Ely, George J. Mailath, and Larry Samuelson. 2008. Common Learning. *Econometrica* 76, 4 (July 2008), 909–933. <https://doi.org/10.1111/j.1468-0262.2008.00862.x>
- Nikhil Devanur, Jason Hartline, Anna Karlin, and Thach Nguyen. 2011. Prior-Independent Multi-parameter Mechanism Design. In *Internet and Network Economics*, Ning Chen, Edith Elkind, and Elias Koutsoupias (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 122–133.
- Jeffrey Ely and Juuso Välimäki. 2003. Bad Reputation. *The Quarterly Journal of Economics* 118, 3 (2003), 785–814. <https://EconPapers.repec.org/RePEc:oup:qjecon:v:118:y:2003:i:3:p:785-814>.
- Jad Esber and Scott Duke Kominers. 2021. A Novel Framework for Reputation-Based Systems. <https://a16zcrypto.com/posts/article/reputation-based-systems/>
- Jean-Louis Foulley, Gilles Celeux, and Julie Josse. 2018. Empirical Bayes approaches to PageRank type algorithms for rating scientific journals. arXiv:1707.09508 [stat.ME] <https://arxiv.org/abs/1707.09508>
- Drew Fudenberg and David K. Levine. 1992. Maintaining a Reputation when Strategies are Imperfectly Observed. *The Review of Economic Studies* 59, 3 (July 1992), 561. <https://doi.org/10.2307/2297864>
- Drew Fudenberg and Yuichi Yamamoto. 2011. Learning from private information in noisy repeated games. *Journal of Economic Theory* 146, 5 (Sept. 2011), 1733–1769. <https://doi.org/10.1016/j.jet.2011.03.003>
- Fangyu Gai, Baosheng Wang, Wenping Deng, and Wei Peng. 2018. Proof of Reputation: A Reputation-Based Consensus Protocol for Peer-to-Peer Network. In *DASFAA*.
- Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. 2013. WTF: the who to follow service at Twitter. In *Proceedings of the 22nd International Conference on World Wide Web (Rio de Janeiro, Brazil) (WWW '13)*. Association for Computing Machinery, New York, NY, USA, 505–514. <https://doi.org/10.1145/2488388.2488433>
- John C. Harsanyi. 1982. *Games with Incomplete Information Played by "Bayesian" Players*. Springer Netherlands, Dordrecht, 154–170. https://doi.org/10.1007/978-94-017-2527-9_8
- Jason D. Hartline and Tim Roughgarden. 2009. Simple versus optimal mechanisms. In *Proceedings of the 10th ACM Conference on Electronic Commerce (Stanford, California, USA) (EC '09)*. Association for Computing Machinery, New York, NY, USA, 225–234. <https://doi.org/10.1145/1566374.1566407>
- Wassily Hoeffding. 1963. Probability Inequalities for Sums of Bounded Random Variables. *J. Amer. Statist. Assoc.* 58, 301 (March 1963), 13–30. <https://doi.org/10.1080/01621459.1963.10500830>
- John Hopcroft and Daniel Sheldon. 2008. Network Reputation Games. *eCommons@Cornell* (2008).
- Olle Häggström. 2002. *Irreducible and aperiodic Markov chains*. Cambridge University Press, 23–27.
- Gábor Iván and Vince Grolmusz. 2010. When the Web meets the cell: using personalized PageRank for analyzing protein interaction networks. *Bioinformatics* 27, 3 (12 2010), 405–407. <https://doi.org/10.1093/bioinformatics/btq680>

- arXiv:https://academic.oup.com/bioinformatics/article-pdf/27/3/405/48865151/bioinformatics_27_3_405.pdf
- Leonard Kleinrock, Rafail Ostrovsky, and Vassilis Zikas. 2020. Proof-of-Reputation Blockchain with Nakamoto Fallback. In *INDOCRYPT (Lecture Notes in Computer Science, Vol. 12578)*. Springer, 16–38.
- Patrick L. Leoni. 2014. Learning in General Games with Nature’s Moves. *Journal of Applied Mathematics* 2014 (2014), 1–9. <https://doi.org/10.1155/2014/453168>
- David A. Levin, Yuval Peres, and Elizabeth L. Wilmer. 2006. *Markov chains and mixing times*. American Mathematical Society. http://scholar.google.com/scholar.bib?q=info:3wf9IU94tyMJ:scholar.google.com/&output=citation&hl=en&as_sdt=2000&ct=citation&cd=0
- J. M. Maestre and H. Ishii. 2016. A cooperative game theory approach to the PageRank problem. In *2016 American Control Conference (ACC)*. 3820–3825. <https://doi.org/10.1109/ACC.2016.7525508>
- G.J. Mailath and L. Samuelson. 2006. *Repeated Games and Reputations: Long-Run Relationships*. Oxford University Press, USA. <https://books.google.com/books?id=hAISDAAAQBAJ>
- Lik Mui. 2002. *Computational models of trust and reputation: agents, evolutionary games, and social networks*. Ph.D. Dissertation. Massachusetts Institute of Technology, Cambridge, MA, USA.
- John Nash. 1951. Non-Cooperative Games. *The Annals of Mathematics* 54, 2 (Sept. 1951), 286. <https://doi.org/10.2307/1969529>
- Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V. Vazirani. 2007. *Algorithmic Game Theory*. Cambridge University Press, USA.
- Jérôme Renault and Tristan Tomala. 2004. Learning the state of nature in repeated games with incomplete information and signals. *Games and Economic Behavior* 47, 1 (2004), 124–156. [https://doi.org/10.1016/S0899-8256\(03\)00153-2](https://doi.org/10.1016/S0899-8256(03)00153-2)
- P. Resnick, R Zechauser, E Friedman, and Ko Kuwabara. 2001. Reputation Systems: Facilitation trust in Internet Interactions. *Journal of Communications - JCM* 43 (01 2001).
- Klaus M Schmidt. 1993. Reputation and Equilibrium Characterization in Repeated Games with Conflicting Interests. *Econometrica* 61, 2 (March 1993), 325–351.
- Takuo Sugaya and Yuichi Yamamoto. 2020. Common learning and cooperation in repeated games. *Theoretical Economics* 15, 3 (2020), 1175–1219. <https://doi.org/10.3982/te3820>
- Cheng Sun. 2015. *REPUTATION GAMES AND POLITICAL ECONOMY*. Ph.D. Dissertation. PRINCETON UNIVERSITY, USA.
- W. Xing and A. Ghorbani. 2004. Weighted PageRank algorithm. In *Proceedings. Second Annual Conference on Communication Networks and Services Research, 2004*. 305–314. <https://doi.org/10.1109/DNSR.2004.1344743>
- Mingji Yang, Hanzhi Wang, Zhewei Wei, Sibao Wang, and Ji-Rong Wen. 2024. Efficient Algorithms for Personalized PageRank Computation: A Survey. *IEEE Transactions on Knowledge & Data Engineering* 36, 09 (Sept. 2024), 4582–4602. <https://doi.org/10.1109/TKDE.2024.3376000>
- J. Yu, D. Kozhaya, J. Decouchant, and P. Esteves-Verissimo. 2019. ReputCoin: Your Reputation Is Your Power. *IEEE Trans. Comput.* 68, 8 (Aug 2019), 1225–1237. <https://doi.org/10.1109/TC.2019.2900648>
- Shmuel Zamir. 2009. *Bayesian Games: Games with Incomplete Information*. Springer New York, New York, NY, 426–441. https://doi.org/10.1007/978-0-387-30440-3_29