MeritRank
Sybil Tolerant Reputation for Merit-based Tokenomics

Nasrulin, Bulat; Ishmaev, Georgy; Pouwelse, Johan

DOI
10.1109/BRAINS55737.2022.9908685

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 2022 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)

2022 4th Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS) | 978-1-6654-7158-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/BRAINS55737.2022.9908685

MeritRank: Sybil Tolerant Reputation for Merit-based Tokenomics

Bulat Nasrulin, Georgy Ishmaev, Johan Pouwelse
Delft University of Technology
{b.nasrulin, m.a.devos-1, g.ishmaev, j.a.pouwelse,}@tudelft.nl

***Abstract*—Decentralized reputation schemes present a promising area of experimentation in blockchain applications. These solutions aim to overcome the shortcomings of simple monetary incentive mechanisms of naive tokenomics. However, there is a significant research gap regarding the limitations and benefits of such solutions. We formulate these trade-offs as a conjecture on the irreconcilability of three desirable properties of the reputation system in this context. Such a system can not be simultaneously generalizable, trustless, and Sybil resistant. To handle the limitations of this trilemma, we propose MeritRank: Sybil tolerant feedback aggregation mechanism for reputation. Instead of preventing Sybil attacks, our approach successfully bounds the benefits of these attacks. Using a dataset of participants' interactions in MakerDAO, we run experiments to demonstrate Sybil tolerance of MeritRank. Decay parameters of reputation in MeritRank: transitivity decay and connectivity decay, allow for a fine-tuning of desirable levels of reputation utility and Sybil tolerance in different use contexts.**

***Index Terms*—Reputation, Sybil attack, Tokenomics, Feedback Aggregation**

I. Introduction

Reputation mechanisms in blockchain applications can provide many desirable properties as system components. In general, reputation mechanisms can be employed at different layers of blockchain systems: protocols layer e.g Delegated Proof-of-Stake [1], middleware e.g. MEV-Boost [2], and most prominently at the application layer of Decentralized Autonomous Organizations (DAOs) [3]–[6]. However, the implementation of reputation-based incentives is still not well researched and rife with challenges. There is a deficit of systematic understanding regarding general trade-offs of different approaches to reputation. The most prominent research gap in this context is the issue of Sybil attacks, a problem well known in peer-to-peer protocols [7]. This problem presents a significant barrier to the implementation of reputation systems proposed as a way to address the limitations of simple tokenomics [8], [9].

Token-based incentives became one of the most prominent mechanisms in blockchain protocols, solving a number of open incentivization problems in peer-to-peer networks, such as network liveness, network security, and open-sources software maintenance. Experimentation with these mechanisms has contributed to the emergence of a subfield in the blockchain design space, sometimes labelled 'tokenomics'. In fact, all prominent application cases for blockchain protocols, including cryptocurrencies, Decentralized Finance (DeFi), DAOs, and Non-Fungible Tokens (NFTs), employ these incentives mechanisms at different levels of abstraction [10].

Limits of such incentives, however, are only slowly becoming apparent with empirical observations. One is a misalignment of incentives that monetary incentives create between different participants in more complex systems [11], [12]. The second limitation is the vulnerability of governance schemes built on naive monetary incentives [13]. Finally, token-based incentives in blockchain protocols disproportionately rewarding large holders tend to encourage re-centralization in decentralized networks [14]. Allocation of rewards for participants[1] of decentralized protocols based on merits or contributions reflected in their non-transferable 'reputation' offers an appealing approach to overcome these limitations. Merit-based rewards can provide alignment of incentives for participants in complex communities and allow for a sustainable distribution of resources [15].

This paper aims to contribute to the theoretical research gap on trade-offs of different reputation solutions and propose a practical, scalable Sybil tolerant reputation mechanism scheme in decentralized networks. We propose and demonstrate feasibility on the experimental basis of MeritRank: *Sybil-tolerant Aggregated Feedback Reputation Mechanism*, compatible with token-based incentive mechanisms. Our contributions are the following:

- We analyze and formulate general trade-offs between desirable properties of reputation in decentralized settings as *decentralized reputation trilemma* in section II.
- We describe the system model for logic and key components for reputation-based distribution of rewards in a generic DAO in section III.
- We present the formalization of MeritRank and three types of reputation decay as a method to achieve tolerance against Sybil attacks to manipulate feedback aggregation reputation in section IV.
- Using experiments on data set from more than 150 weeks of user interactions in MakerDAO, we demonstrate that transitivity decay and connectivity decay provide high tolerance against Sybil attacks on reputation. Another experimental finding is that epoch decay, implemented in other reputation mechanisms [6] worsens Sybil tolerance (section V).

This work was funded by NWO/TKI grant BLOCK.2019.004

[1] We use terms: *participant, node* and *peer* interchangeably.

978-1-6654-7158-9/22/$31.00 ©2022 IEEE

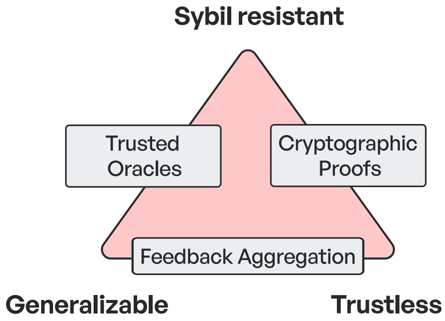


Figure 1: Reputation mechanism trade-off triangle. A reputation mechanism can be only one of the two: Generalizable, Sybil resistant, Trustless.

II. Background and Related Work

Reputation systems in decentralized environments have been proposed for various application-specific cases, and as general models in peer-to-peer systems [16]–[21]. Accordingly, the limitations of these solutions are relatively well-understood. Some of these limitations include scalability [19], contextual accuracy [18], partial reliance on trusted-setups [18], [19], vulnerability to certain types of attacks [22]–[26], and significant privacy trade-offs [20]. However, there is also an identifiable research gap regarding general trade-offs inherent to any reputation systems implemented in decentralized environments. We identify only a small number of surveys on reputation solutions in decentralized settings [18], [19], with only limited comparative trade-off analysis [20]. The absence of engineering research on reputation in the context of tokenomics and DAOs is characterized by the general deficit of academic research in this area [27].

A. Decentralized Reputation Trilemma

We formulate these trade-offs as a conjecture on the irreconcilability of three desirable properties of the reputation system in this context. Such a system can not be simultaneously *generalizable, trustless*, or *Sybil resistant*. The trustless property means that reputation accounting and evaluation does not rely on specific trusted entities. In a fully Sybil-resistant reputation system, the attacker cannot unfairly manipulate the reputation system by controlling multiple identities [28]. Generalizable reputation system allows to account and evaluate any type of participants' contribution.

All proposed reputation mechanisms inevitably sacrifice one of these properties to achieve two others, thus forming a *Decentralized Reputation Trilemma* as illustrated in figure 1. To illustrate this problem, we look into three conceptually different methods to implement reputation.

Trusted Oracles. This reputation method is based on some trusted oracle keeping track of reputation scores. In the context of a decentralized system, such an oracle can take input from the actions of participants to calculate their reputation scores based on a pre-determined reputation function. This is a generalizable approach as oracles can take any input for reputation function [19]. The problem of Sybil resistance can be addressed with a task of Sybil detection outsourced to an oracle verifying the identities of participants [29].

Participants have to trust that oracles use correct input and provide honest outputs on reputation function, which undermines the trustless properties of a decentralized system. Furthermore, dependency on oracles introduces points of failure in a system that can be exploited, as demonstrated by price oracles in DeFi [30]. Attempts to provide trustless properties to oracles essentially create a peer prediction market. However, this reduces their applicability to a set of cases where agreement on reputation can be achieved through social consensus voting, thus sacrificing generalizability [31].

Cryptographic Proofs. Another popular tool in reputation accounting is a method of cryptographic proofs, that can be employed to address the problem of Sybil attacks. Each peer performs some contribution or work verifiable by any other peer in the network. Peers keep track of their and others' reputation scores, generating proofs of contributions and sending them to all other peers in the network. After validating the proofs, clients update their reputation scores. A Sybil attack is undermined by the requirement to provide verifiable proof of contribution. Such a solution can be trustless and decentralized since no single party is responsible for reputation accounting.

However, this approach also has significant limitations regarding generalizability of application. Certain types of contributions like computation work, proof of bandwidth, proof of transaction, or proof of storage allow for accounting mechanisms based on cryptographic proofs, where contribution accounting can be trustless and universally verifiable. However, this type of accounting does not grasp semantic richness of all types of collaborative work and human contributions. Furthermore, even in those application cases where such type of accounting is conceptually meaningful, practical scalability can be hampered by the high overhead [32].

Feedback Aggregation. A peer-to-peer reputation is an approach where participants interact with each other, each giving feedback on the contribution or work provided by others. The resulting reputation is computed by aggregating feedback from participants.

The main benefits of this approach are generalizability and the absence of trusted third parties. Peer-to-peer feedback makes such an approach feasible for various application-specific contexts, as peers can provide context-specific feedback on any interaction or work. Unlike the cryptographic proofs approach, feedback aggregation does not introduce high costs from overhead. However, the big disadvantage of this approach is that it is susceptible to manipulation, especially through Sybil attacks [28].

B. Sybil tolerant feedback aggregation

The observations based on the reputation trilemma necessitate a solution for reputation systems in decentralized environments that do not sacrifice completely one of the corners of this triangle. Hard scalability limitations for *Cryptographic Proofs*,

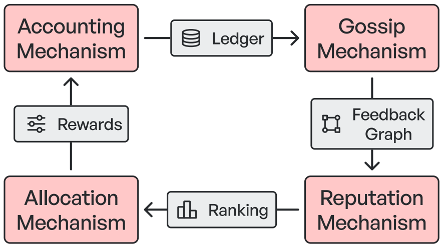


Figure 2: Merit-Based Tokenomics system model

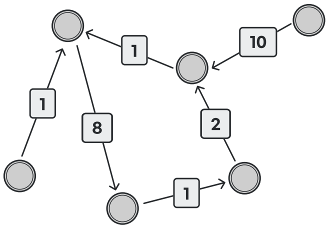


Figure 3: Example of feedback graph G. Edge represent the sum of evaluations made by the participant about another participant.

and fundamental nature of limitations presented by *Trusted Oracles* suggest that the *Feedback Aggregation* side presents the most feasible direction to achieve acceptable trade-offs.

In the context of open, permissionless systems, strict Sybil resistance is not achievable entirely [33]. Furthermore, attempts to emulate Sybil resistant properties of closed systems undermine the privacy of participants and other desirable properties of decentralized solutions. Different approaches can be used to minimize the impact of Sybil attacks on peer feedback-based reputation: *Sybil detection* and *Sybil tolerance* [34], [35]. The efficiency of Sybil detection, in general, is limited by the assumptions on the behavior of Sybil nodes that is distinctively different from the behavior of honest nodes in the network, which is not necessarily always the case [33].

Sybil tolerance for aggregated feedback reputation mechanisms is a system property evaluated in the context of specific Sybil attacks on reputation. Sybil tolerance does not try to identify and filter Sybil identities but instead limits their impact on the system. This approach allows for straightforward application in those cases where impact can be evaluated in consumed system resources, e.g., rewards pool in a DAO treasury. We also demonstrate in Section V that two properties of this approach suggest its feasibility. (1) Tolerance threshold remains constant with the increase in the size of Sybil participants. (2) Threshold can be adjusted for different fairness and accuracy levels depending on reputation's contextual semantics.

III. Merit-based Tokenomics

In this section, we present a system model for merit-based tokenomics. This model is generic and does not make assumptions about implementation details. It can describe a reputation system for the participants of a generic DAO, where peers provide feedback to each other resulting in a reputation ranking that can be used to distribute rewards from DAO treasury proportionally to accrued reputation. Figure 2 represents the logic and key components of the system model.

A. System Model

Accounting mechanism. We assume that peers observe and evaluate each others' contributions. The evaluation is recorded in a personal ledger. We model the interaction between peers as a directed graph $G = (V, E, w)$, called *the feedback graph*. An edge $(i, j) \in E$ is directed from i to j and total feedback given to j by i. We model the feedback as a special token value, which accumulates with time. Peers perform some work contributing to the community; as a result, they receive feedback from other peers. The function $w : V \times V \to \mathbb{R}_{\geq 0}$ denotes the weight of the edges and shows the total feedback received.

An example of a feedback graph is presented in Figure 3. The weights of the edges are cumulative weighted feedback from participants. Our model is agnostic to the way the feedback graph is formed or how the graph weights are chosen. Specific semantics of feedback are left out of the scope of the model. Some examples may include code contributions, governance proposals, or a rating assigned to j based on the reactions of i on Github [36]. An accounting mechanism can be implemented with middleware that defines feedback semantics and records evaluations in a personal ledger, similar to SourceCred [5].

Gossip mechanism. We assume that there is a way for the peer to discover the feedback graph, for example, through a gossip protocol. While we acknowledge that reputation mechanisms in distributed systems face problems, such as the incompleteness of information about peer interactions, peer discovery, etc., we disregard these issues assuming that global information is available through a fault-free gossip protocol. Practical implementation may use overlay for peer-to-peer gossip similar to [37].

Reputation mechanism. A reputation mechanism assigns some reputation scores for every known node in the feedback graph. In principle, a reputation score reflects the level of contributions made by the node compared with others.

Definition III.1 (Reputation score)**.** A reputation score $R(G, j)$ is a value assigned to node j given known feedback graph G:

$$R(G, j) \in \mathbb{R} \quad \forall j \in V$$

We refer to $R(G)$ as a set of reputation scores for all nodes in the graph G. The reputation score is computed by aggregating the feedback from all nodes in the graph.

Allocation mechanism. Reputation mechanisms are used as an input for some allocation policy that decides, given a set of nodes to whom distribute the reward. For model simplicity, it is assumed that allocation happens automatically per each epoch without participants' actions. Participants may claim tokens from the reward pool for each epoch in practical implementation, corresponding to their reputation at the end of each epoch.

Definition III.2 (Allocation policy)**.** An allocation policy A returns rewards given the feedback graph G and reputation scores $R_i(G)$:

$$A_i(R_i(G_i), j) \in \mathbb{R}$$

We do not make any assumptions about the allocation policies but assume that nodes holding a high reputation score are more likely to receive rewards or receive higher rewards. One example of an allocation is *winners-take-all*, in which the top ten nodes with the highest reputation receive all the rewards. Another example of allocation policy is the quadratic distribution [9], in which nodes receive proportionally less with the increase in reputation. Different policies will have different application fit, but discussion on specific properties is out-of-scope.

Dynamic model. To capture the diachronic nature of contributions and rewards we consider *epochs*. This reflects the semantics of reputation which accumulates over time. One epoch represents one full cycle of system model shown in Figure 2. We use superscript to refer to the epoch. For example, $G^{(\tau)}$, for the feedback graph at epoch $\tau \geq 0$.

B. Sybil Tolerance Model

We model a Sybil attack as a *strategic*, in a sense that the attacker is interested in receiving as many rewards with only a small amount of work. It does this by creating fake identities and fake edges connecting identities it controls with one another. The weights of the edges in the Sybil region can be chosen arbitrarily by the attacker. We assume that the attacker knows which reputation algorithm is used and can execute an optimal attack on the given reputation mechanism.

Definition III.3 (Sybil Attack)**.** Given the feedback graph $G = (V, E, w)$, an attacker j performs a Sybil attack σ_S by introducing the following elements to the graph:

- A set of Sybil identities $S = \{j, s_1, \ldots, s_m\}$, each of which is called a **Sybil** and is indistinguishable from an honest node by other nodes.
- A set of **Sybil edges** $E_S \subset S \times S$ with edge weights $w_S : S \times S \to \mathbb{R}_{\geq 0}$.
- A set of **attack edges** E_a with weights $w_a : V \times S \to \mathbb{R}_{\geq 0}$.

After the attack has been carried out, we obtain a modified feedback graph, denoted $G' := G \downarrow \sigma_S$.

Sybil attack benefit. The attacker is allowed to create arbitrarily many Sybil nodes and fake edges between Sybils. However, we require any edge between honest nodes V and Sybils S to be a real transaction. Therefore, we assume that the attacker also makes contributions to receive feedback from honest nodes in order to create an edge connecting the attacker's identities to the honest part of the network. To maximize the benefit of such, the attacker might decide to get feedback from a highly reputable node via its attack edge.

Due to dynamic nature of allocation we use reputation scores as proxies for cost and profit of the Sybil attack. The profit for the Sybil attack σ_S given the modified feedback graph G' is defined as:

$$\omega^+(\sigma_S) = \sum_{s \in S \setminus \{j\}} R(G', s)$$

The cost for the Sybil attack is defined as reputation gained through *honest work* through attack edges E_a. Given an modified graph $G'' = (V \cup S, E \cup E_a, w)$ the cost is defined of the Sybil attack is defined as:

$$\omega^-(\sigma_S) = \sum_{s \in S} R(G'', s)$$

We model the Sybil-Tolerance as a bound on the benefit that the attacker can gain through a Sybil attack σ_S on feedback graph G. A reputation mechanism R is *Sybil tolerant* if the gain after performing a Sybil attack is limited by some constant $c \geq 0$:

$$\lim_{|S| \to \infty} \frac{\omega^+(\sigma_S)}{\omega^-(\sigma_S)} \leq c$$

IV. MeritRank: Sybil Tolerant Reputation Mechanism

This section introduces a generic Sybil tolerant modification to the flow and walk-based reputation mechanisms. We briefly demonstrate the limits to Sybil tolerance for these types of reputation mechanisms. We propose three modifications that limit the gain from a Sybil attack.

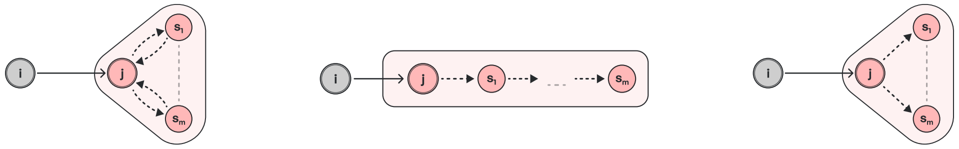
A. Sybil Tolerance of Existing Reputations

We consider reputation mechanisms that satisfy the following requirements: 1.) The reputation mechanism should be a graph-based measure of centrality, 2.) It should work without requiring any central third party, and lastly, 3.) It should reward contributors.

Naive approaches that are based on simple global centralities, such as vertex degrees, are easily manipulable. In fact, even complex global reputation mechanisms, such as PageRank, are not Sybil tolerant [22]. Such vulnerability trivially follows from the fact that any node in the network can gain a reputation by adding any new edge. Thus, the attacker can unboundedly gain benefits from the Sybil attack without the need to create any attack edge.

To address this issue, personalized reputation mechanisms were proposed [38]. A personalized reputation mechanism assigns a positive reputation to the node j only if there is a path from a seed node i to the node j. This, in turn, provides inherent tolerance to a simple Sybil attacks, as participants need to perform some work to establish a path.

We analyze three generic reputation mechanisms previously reported to be Sybil tolerant. Specifically, we consider MaxFlow [39], personalized PageRank [40], personalized Hitting Time [38]. We, however, show that these reputation mechanisms are vulnerable to the Sybil attack as defined in III.3. To illustrate their vulnerabilities, we present three Sybil



(a) Cycle attack

(b) Serial attack

(c) Parallel attack

Figure 4: Sybil attack strategies. A beneficial Sybil attack is a combination of these three strategies.

strategies in Figure 4. These strategies, when combined, cover all Sybil beneficial strategies as shown in [21].

Personalized PageRank: The personalized PageRank algorithm (PPR) with source node i and reset probability $\alpha \in [0, 1]$ on feedback graph G_i is given by the steady-state probability that an α-terminating random walk initiated at i arrives at j. In practice, personalized PageRank is implemented as a random walk-based reputation, counting the number of encounters in a random walk.

The Sybil-Tolerance of PPR follows from its damping factor α, which comes at the cost of reputation's accuracy. The most effective way to attack PageRank is to use cycles with big weights. The attack is shown in the Figure 4a. The cycle attack goes as follows: The attacker inserts one Sybil identity and adds two edges forming a cycle between the Sybil through its Sybils and, as a result, increasing the total number of encounters in a random walk.

Personalized Hitting Time: Given a feedback graph G, the personalized Hitting Time (PHT) algorithm is given by the probability of an α-terminating random walk that starts at i visiting node j before it terminates. As with the PageRank, PHTs are implemented as random-walk reputation. The difference is that it accounts only for facts of node encounters in a random walk.

PHT bounds the gains from both cycle attacks and parallel attacks. However is still vulnerable to the serial attack (Figure 4b). The attacker can create a long path connecting all Sybil nodes. As a result, all Sybil nodes in the path can have a positive reputation.

MaxFlow: Given a seed node i, MaxFlow (MFW) reputation score of node $j \in V_i$ is given by the maximum flow of contributions from j to i in the feedback graph G. MaxFlow takes into account all possible flows of work from the source node to the target node. The amount a node can gain is bounded by its aggregated contributions.

MaxFlow is vulnerable to a parallel attack shown in Figure 4c. The attacker can create multiple Sybil identities $s_1, ..., s_m$, with each identity gaining the same amount of reputation, i.e., for all $k = 1..m | R(G, s_k) = R(G, s_1)$.

B. Bounding the Attacks

We propose three modifications to the above-mentioned reputation mechanisms to improve their Sybil tolerance. To address the vulnerabilities, we require to provide bounds on parallel reports and serial reports and provide bounds on transitivity. Formally these bounds are defined as:

- **Parallel report bound**. Let the attacker execute a parallel attack or cycle attack, as shown in Figure 4, adding at each time one Sybil identity. A reputation R is parallel report bound if $\sum_{l=1}^{m} R(G^{(m)}, s_l) \leq R(G^{(1)}, s_1)$.
- **Serial report bound**. Reputation R is serial-bound if there exists a bound on the gain after a serial attack (as shown in Figure 4b): $\sum_{l=1}^{m} R(G^{(m)}, s_l) < \infty$.
- **Bounded Transitivity**. Let node j received some positive reputation $R(G, j)$ because of the existence of the path $P(i, j)$. The reputation R is transitive-bound if $R(G, j) \leq \min_{k \in P(i,j)} (R(G, k))$.

Relative feedback. To achieve parallel report bound reputation each edge weight must be considered with respect to the degree of the node. This can be achieved by modifying each weight in the graph G, such that, $\bar{w}(i, j) = w(i, j) / \sum_{k \in \mathcal{N}(i)} (w(i, k))$, where $\mathcal{N}(i)$ is the set of neighbours of node i in the graph G.

Transitivity α decay. To achieve serial report bounds, we introduce transitivity decays. For random walk-based reputation, we terminate the random walks at each new step with probability α. The value of α limits the length of random walks, limiting the effects of serial attacks.

Connectivity β decay. We increase Sybil tolerance even further by introducing punishment for being in a separate connected component. The intuition behind this decay is that the attack edges, as defined in definition III.3 are often bridges, i.e., their cut creates two separate components.

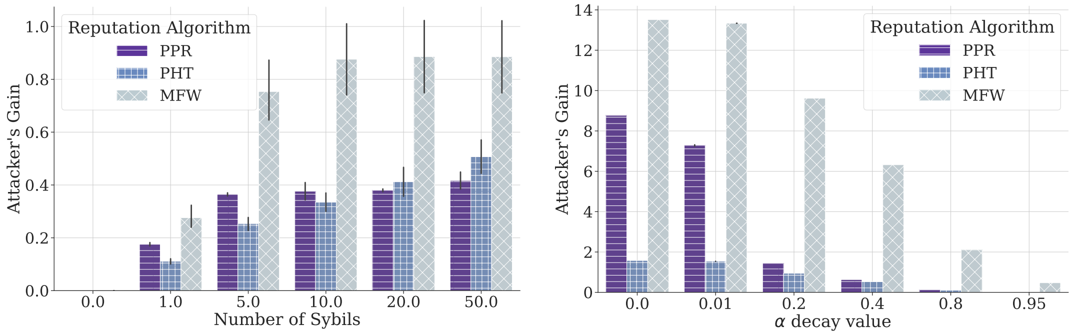
We punish the nodes for forming a separate component with a decay multiplier β. Given the feedback graph G and seed node i, the modified reputation for node j is defined as follows:

$$R_\beta(G, j) = \begin{cases} (1 - \beta) * R(G, j), & \text{if } I_i(j) \\ R(G, j), & \text{otherwise} \end{cases} \tag{1}$$

, where $I(i, j)$ is a index indicating that nodes i and j are connected through a bridge.

In practice, the index $I(i, j)$ is calculated based on the proportions of random walks T_{ij} starting from a seed node i that reach node j through some node k in a path $P(i, j)$. Specifically, k is a cut vertex if the proportion of random walks is higher that some given threshold $t > 0$:

$$I_i(j) = \frac{\max_{k \in P(i,j)} |T_{ij}(k)|}{|T_{ij}|} \geq t$$



(a) Gain for given reputation algorithms with $\alpha = 0.4$ (b) Attacker's gain with given α decay with $|S| = 50$.

Figure 5: The total reputation gain by adding Sybil nodes with transitivity α decay.

If the threshold is equal to 1, this would indicate that vertex k is the only one connecting i and j. The higher the connectivity decay β, the more important the diversity of feedback received.

Epoch γ decay. Finally, we consider an epoch decay γ. The attacker can gain an additional advantage by reusing an old connection. For example, the attacker can create a few attack edges and then, as a result, gain a positive reputation for a prolonged period of time.

We provide two implementations of this decay: (1) when the multiplier is applied to reputation values, for i.e., $R_\gamma(G^{(\tau+1)}, i) = (1 - \gamma) * R_\gamma(G^{(\tau)}, i) + R(G^{(\Delta(\tau+1,\tau))}, i)$, where $G^{(\Delta(\tau+1,\tau))} = (V^{(\tau+1)}, E^{(\tau+1)} \setminus E^{(\tau)}, w^{(\tau+1)})$, and (2) when the decay is applied to graph weights, i.e., $w_\gamma^{(\tau+1)}(i,j) = w^{(\tau+1)}(i,j) - (2 - \gamma) * w^{(\tau)}(i,j)$.

V. Experiments

In this section, we show the result of a quantitative study of the attacker manipulation of reputation by simulating multiple Sybil attacks using real-world dataset. We show effects of different decay values on different reputation algorithms and resulting thresholds for Sybil tolerance.

A. Experimental Set-Up

We create a dataset from the forum of one of the biggest DAOs to this date - MakerDAO. We parse all user interactions, replies, likes, created posts, and votes on the proposals. The users' actions are recorded as contributions to the DAO weighted in work units. Each participant is rewarded with DAI token proportional to the reputation at each epoch. Thus the attacker gain is directly expressed in DAI tokens.

We use weights as defined in SourceCred MakerDAO project[2]. For instance, a post created by a user is evaluated by the number of likes it receives. One like of the post is equal to 4 work units. The contribution graph is a user-to-user evaluation, where the weight of an edge (i,j) is the total sum of work done by j as evaluated by i. Our dataset [3] contains activity starting from Jun 24, 2019, to May 26, 2022. We use the epoch of one week. The resulting graph $G^{(153)}$ contains 2057 nodes and 35853 edges. The peak week 149 has data of 291 users and 1528 edges.

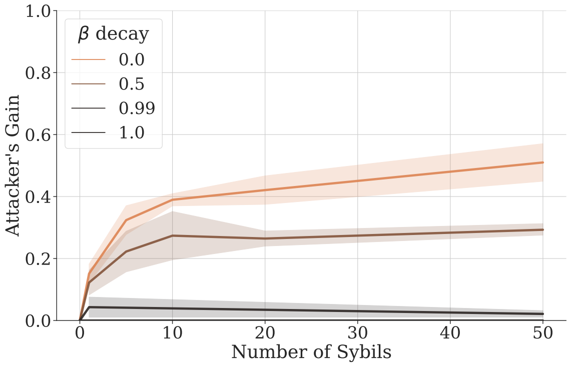


Figure 6: Attacker's gain with β connectivity decay. We fix $\alpha = 0.4$.

For our computations, we use a special virtual seed node inserted to the graph. At each epoch, we connect the seed node with the top 10 most reputable nodes with equal weights.

We perform the Sybil attack as described in definition III.3. We implement all attack strategies as shown in Figure 4. We pick the attacker node j randomly from the set of 10 most reputable nodes. The attack is performed repeatedly every epoch after a period of 20 epochs. We argue that this attack have the most prominent effects on reputation.

B. Transitivity α Decay

We apply the transitivity decay to personalized PageRank (PPR), Hitting Time (PHT), and MaxFlow (MFW). We show that it successfully limits the attacker gain in Figure 5a. Note that random walk-based reputation achieves higher tolerance than flow-based reputation.

We report the attack again with the increase of Sybils in Figure 5b. Transitivity decay can successfully bound the number of the gain of the attacker. The effect is most visible with higher numbers of α. A downside, however, is the potential loss of reputation utility, as shown in section V-E.

[2] http://makerdao.sourcecred.io

[3] We release source code and the dataset at https://github.com/grimadas/meritrank

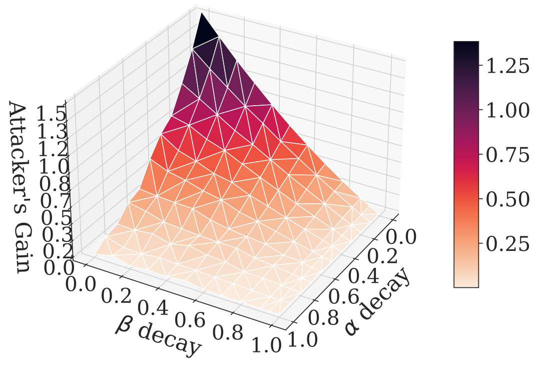
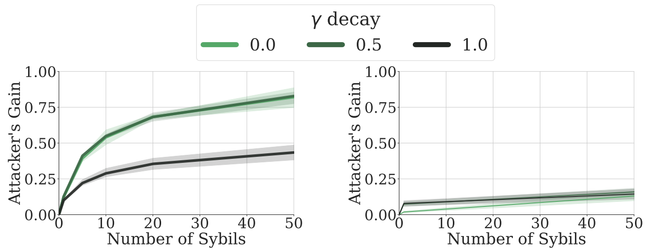


Figure 7: Cumulative effect of both α and β decay mechanisms with $|S|= 50$.



(a) γ decay applied to reputation values.

(b) γ decay applied to graph weights.

Figure 8: The total reputation gain of the attacker with γ-epoch decay. We fix $\alpha = 0.4$ and $\beta = 0.5$.

C. Connectivity β Decay

We report the effects of connectivity decay on the attacker's gain in Figure 6. We fix the transitivity decay $\alpha = 0.4$. The connectivity decay mechanism can further improve the Sybil tolerance of transitive decay. Note that with $\beta = 0.0$, we are able to eliminate the gains of the attacker fully.

We show in Figure 7 that connectivity and transitivity decay combined provided a sufficient level of Sybil tolerance. The figure shows the effect of decays on the attacker's gain. Two decay mechanisms complement each other, providing higher level of Sybil tolerance.

D. Epoch γ Decay

We run the experiments by applying the γ epoch decay to reputation and the weights. Surprisingly, epoch decay worsens Sybil-Tolerance of the reputation mechanism, as can be seen from Figure 8a and Figure 8b. Attacker gains the minimum when there are no epoch decays, i.e., $\gamma = 0.0$. With the increase of γ, the attacker is able to gain more from the Sybil attack. This effect is explained by the fact that reputation weights are higher for new feedback. However, as the attacker can always insert new Sybil identities at each epoch, eventually, these identities receive more walks than honest nodes.

E. Effect of Decays on Informativeness

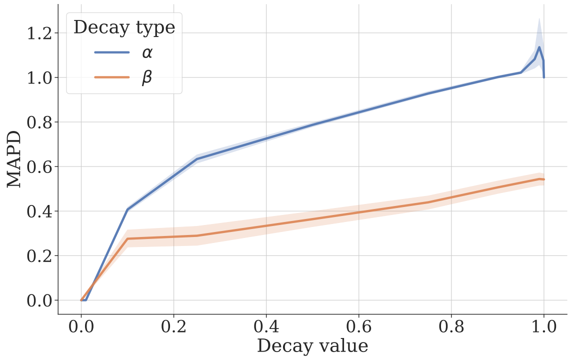


Figure 9: Mean absolute percentage deviation (MAPD) compared with decay value 0 for transitivity decay (α) and connectivity decay (β).

Informativeness of reputation is an application-specific metric that shows the utility and goals of the reputation mechanism. As a reference, we propose to measure the informativeness loss by comparing it with the reputation without the decay. Specifically, we use Mean Absolute Percentage Deviation (MAPD) for decay type and decay value d defined as:

$$MAPD(d) = \frac{1}{|V|}\sum_{j \in V} |\frac{R_{decay=0}(G,j) - R_{decay=d}(G,j)}{R_{decay=0}(G,j)}|$$

We show the effects of decays on the informativeness of the reputation mechanism in Figure 9. The less is the value of MAPD; the less is the effect of decay on reputation informativeness. As expected, connectivity β decay can achieve higher informativeness compared to transitivity decay α. Note that a small increase of MAPD is expected as the decay value get's closer to 1. This happens because first-hop neighbors get a higher reputation than more distant nodes, which are not reachable by random walks.

F. Discussion

Our experiments show that flow-based mechanisms can be translated from credit networks [34] to feedback graphs to achieve Sybil tolerance. However, random walk-based mechanisms PPR and PHT provide better tolerance than flow-based MFW in feedback-based reputation mechanisms. Connectivity decay is preferable as it allows for minimal informativeness loss while providing the highest tolerance than transitivity decay. In practice, both of these parametrizations can be combined for a desirable threshold tolerance.

Our findings also suggest that the intuitively appealing concept of time-based epoch-based decay [6] should be implemented with caution. When applied naively, Reputation mechanism with this decays favor new feedback and feedback from new users assigning higher weights improving gains from dynamic Sybil attack.

VI. Conclusion

The advances in complex blockchain applications reveal limits of naive tokenomics based on simple models of monetary incentives for token holders. Merit-based non-transferable reputation schemes that reward active contributors present a promising direction of engineering for novel blockchain applications such as DAOs. However, application of reputation in decentralized environments is limited by the *reputation trilemma*. We argue that the feedback aggregation mechanism is the most feasible approach for such solutions as it does not sacrifice the trustless properties of decentralization. While some existing solutions, such as SourceCred address the problem of feedback accounting, no solutions are providing Sybil tolerance for reputation mechanisms based on feedback aggregation.

We propose MeritRank: a Sybil tolerant reputation mechanism based on feedback aggregation from the participants of decentralized applications. We use data set from interactions of MakerDAO participants to show experimentally that MeritRank reputation mechanism bounds the profit of attacker using Sybils. The novelty of this approach lies with modifications for reputation decay: transitivity decay, epoch decay, and connectivity decay. Experimental results show that a combination of transitivity decay and connectivity decay can provide a desirable level of Sybil tolerance. Another advantage of our approach is the ability to fine-tune the parameters in order to achieve the context-specific balance between the utility of reputation and Sybil tolerance. These properties suggest that MeritRank provides practical solution to the reputation trilemma.

References

[1] C. T. Nguyen *et al.*, “Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities,” *IEEE Access*, vol. 7, 2019.

[2] MEV-boost: Merge ready flashbots architecture. [Online]. Available: https://ethresear.ch/t/mev-boost-merge-ready-flashbots-architecture/11177

[3] AragonDAO, “Reputation template.” [Online]. Available: https://documentation.aragon.org/products/aragon-client/how-to-create-a-dao-using-aragon-client/page-1

[4] “Coordinape.” [Online]. Available: https://github.com/coordinape

[5] “SourceCred.” [Online]. Available: https://github.com/sourcecred

[6] A. Rea *et al.*, “Colony. technical white paper.” [Online]. Available: https://colony.io/whitepaper.pdf

[7] J. Dinger and H. Hartenstein, “Defending the sybil attack in p2p networks: Taxonomy, challenges, and a proposal for self-registration,” in *First International Conference on Availability, Reliability and Security (ARES’06)*. IEEE, 2006.

[8] P. De Filippi *et al.*, “Reputation,” *Internet Policy Review*, vol. 10, no. 2, 2021. [Online]. Available: https://policyreview.info/glossary/reputation

[9] E. Weyl, P. Ohlhaver, and V. Buterin, “Decentralized society: Finding web3’s soul.” [Online]. Available: https://ssrn.com/abstract=4105763

[10] P. Freni, E. Ferro, and R. Moncada, “Tokenization and blockchain tokens classification: a morphological framework,” in *2020 IEEE Symposium on Computers and Communications (ISCC)*. IEEE, 2020.

[11] P. Garg. MakerDAO whale with 94% voting power reduces dai stability fee by 4%. [Online]. Available: https://cryptoslate.com/makerdao-whale-with-94-voting-power-reduces-dai-stability-fee-by-4/

[12] K. Liam J. How the juno network DAO voted to revoke a whale’s tokens. [Online]. Available: https://decrypt.co/95435/juno-network-dao-proposal-16-voted-to-revoke-tokens-from-whale

[13] P. Daian *et al.* On-chain vote buying and the rise of dark DAOs. [Online]. Available: https://hackingdistributed.com/2018/07/02/on-chain-vote-buying/

[14] S. Martinazzi and A. Flori, “The evolving topology of the lightning network: Centralization, efficiency, robustness, synchronization, and anonymity,” *Plos one*, vol. 15, no. 1, 2020.

[15] M. A. Nowak, “Five rules for the evolution of cooperation,” *Science*, vol. 314, no. 5805, pp. 1560–1563, 2006.

[16] K. Walsh and E. G. Sirer, “Experience with an object reputation system for peer-to-peer filesharing.” in *NSDI*, vol. 6, 2006.

[17] R. Delaviz *et al.*, “Sybilres: A sybil-resilient flow-based decentralized reputation mechanism,” in *2012 IEEE 32nd International Conference on Distributed Computing Systems*. IEEE, 2012.

[18] F. Hendrikx *et al.*, “Reputation systems: A survey and taxonomy,” *Journal of Parallel and Distributed Computing*, vol. 75, 2015.

[19] E. Bellini *et al.*, “Blockchain-based distributed trust and reputation management systems: A survey,” *IEEE Access*, vol. 8, 2020.

[20] S. Gurtler and I. Goldberg, “SoK: Privacy-preserving reputation systems,” *Proceedings on Privacy Enhancing Technologies*, no. 1, 2021.

[21] A. Stannat *et al.*, “Achieving sybil-proofness in distributed work systems,” in *International Conference on Autonomous Agents and Multiagent Systems*, 2021.

[22] A. Cheng and E. Friedman, “Sybilproof reputation mechanisms,” in *Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems - P2PECON ’05*. ACM Press, 2005.

[23] K. Hoffman *et al.*, “A survey of attack and defense techniques for reputation systems,” *ACM Computing Surveys (CSUR)*, 2009.

[24] B. Viswanath *et al.*, “Exploring the design space of social network-based sybil defenses,” in *COMSNETS*. IEEE, 2012.

[25] E. Koutrouli and A. Tsalgatidou, “Taxonomy of attacks and defense mechanisms in p2p reputation systems—lessons for reputation system designers,” *Computer Science Review*, vol. 6, no. 2, 2012.

[26] S. Seuken *et al.*, “Work accounting mechanisms: Theory and practice,” in *Working Paper. Department of Informatics*. University of Zurich, 2014.

[27] Y. El Faqir *et al.*, “An overview of decentralized autonomous organizations on the blockchain,” in *Proceedings of the 16th International Symposium on Open Collaboration*. ACM, 2020.

[28] J. R. Douceur, “The sybil attack,” in *International workshop on peer-to-peer systems*, 2002.

[29] D. Siddarth *et al.*, “Who watches the watchmen? a review of subjective approaches for sybil-resistance in proof of personhood protocols,” *Frontiers in Blockchain*, vol. 3, 2020.

[30] S. Eskandari *et al.*, “SoK: oracles from the ground truth to market manipulation,” in *Proceedings of the 3rd ACM Conference on Advances in Financial Technologies*, 2021.

[31] Y. Cai *et al.*, “A truth-inducing sybil resistant decentralized blockchain oracle,” in *2020 2nd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS)*. IEEE, 2020.

[32] K. Wüst and A. Gervais, “Do you need a blockchain?” in *2018 Crypto Valley Conference on Blockchain Technology (CVCBT)*. IEEE, 2018, pp. 45–54.

[33] L. Alvisi *et al.*, “Sok: The evolution of sybil defense via social networks,” in *IEEE Symposium on Security and Privacy*. IEEE, 2013.

[34] B. Viswanath, M. Mondal, K. P. Gummadi, A. Mislove, and A. Post, “Canal: Scaling social network-based sybil tolerance schemes,” in *Proceedings of the 7th ACM european conference on Computer Systems*, 2012.

[35] B. Viswanath *et al.*, “Exploring the design space of social network-based sybil defenses,” in *2012 Fourth International Conference on Communication Systems and Networks (COMSNETS 2012)*. IEEE, 2012.

[36] E. Miyazono, “Sourcecred: An introduction to calculating cred and grain,” Mar 2020. [Online]. Available: https://research.protocol.ai/blog/2020/sourcecred-an-introduction-to-calculating-cred-and-grain/

[37] D. Tarr *et al.*, “Secure scuttlebutt: An identity-centric protocol for subjective and decentralized applications,” in *Proceedings of the 6th ACM Conference on Information-Centric Networking*. ACM, 2019.

[38] B. K. Liu *et al.*, “Personalized hitting time for informative trust mechanisms despite sybils,” in *Proceedings of the International Conference on Autonomous Agents & Multiagent Systems*, 2016.

[39] M. Meulpolder *et al.*, “Bartercast: A practical approach to prevent lazy freeriding in p2p networks,” in *IEEE International Symposium on Parallel & Distributed Processing*, 2009.

[40] B. Bahmani *et al.*, “Fast incremental and personalized pagerank,” *VLDB*, 2010.