Sybil-proof Accounting Mechanisms with Transitive Trust

Sven Seuken
Department of Informatics
University of Zurich
seuken@ifi.uzh.ch

David C. Parkes
School of Engineering & Applied Sciences
Harvard University
parkes@eecs.harvard.edu

ABSTRACT

For the design of distributed work systems like P2P file-sharing networks it is essential to provide incentives for agents to work for each other rather than free ride. Several mechanisms have been proposed to achieve this goal, including currency systems, credit networks, and accounting mechanisms. It has proven particularly challenging to provide *robustness to sybil attacks*, i.e., attacks where an agent creates and controls multiple false identities. In this paper, we consider *accounting mechanisms* for domains in which (1) transactions cannot be bound to reports, (2) transactions are bilateral and private, and (3) agents can only form trust links upon successful work interactions. Our results reveal the trade-off one must make in designing such mechanisms. We show that accounting mechanisms with a strong form of transitive trust cannot be robust against strongly beneficial sybil attacks. However, we also present a mechanism that strikes a balance, providing a weaker form of transitive trust while also being robust against the strongest form of sybil attacks. On the one hand, our results highlight the role of strong social ties in providing robustness against sybil attacks (such as those leveraged in credit networks using bilateral IOUs), and on the other hand our results show what kind of robustness properties are possible and impossible in domains where such pre-existing trust relations do not exist.

Categories and Subject Descriptors

J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

Keywords

Sybil-proofness; Transitive Trust; Peer-to-Peer; Mechanism Design; Credit Networks

1. INTRODUCTION

Distributed work systems arise in many places, for example in peer-to-peer file-sharing networks, in decentralized ride-sharing systems, and in *ad hoc* wireless networks where individual peers route data packages for each other. A central goal in such systems is to enable useful transactions between agents, while providing incentives for them to perform work rather than free-ride. A common theme is finding ways to avoid the *double coincidence of wants* problem: it is almost never the case that two agents can both provide useful work for each other *and* are at the same time interested in the work provided by each other. Rather, it is desirable to allow one agent to work for another agent now, receiving work from the same or some other agent at a later point in time.

However, allowing such temporally disconnected, one-sided work interactions can lead to an incentive problem in the absence of trust: strategic agents may consume work from others, but never reciprocate. Even systems with a globally trusted currency, including micro payment systems and virtual currencies like BitCoin [16], can suffer from free-riding in the absence of the ability to bind a payment to a work transaction. In particular, the party that acts first must *trust* the other party to cooperate later.

1.1 Transitive Trust and Sybil-Proofness

Thus, trust is important to enable economic transactions. However, requiring *direct* trust links for all transactions is very limiting because in many domains, two agents may want to interact that have never met before. In such situations, intermediaries can help if we allow for *transitive trust*: if agent A trusts B and B trusts C, then A could decide to also trust C, at least to some degree. This notion of transitive trust is very natural in many domains, e.g., employees recommending new employees or doctors recommending other doctors. In large networks, where the rendezvous-probability of two agents is small, transitive trust is essential, increasing the efficiency of the system by enabling many valuable transactions.

Intuitively, allowing for transitive trust opens up more possibilities for sybil attacks, where an agent creates and controls multiple false identities. For example, a trusted agent might claim that its sybils are trustworthy and then let the sybils consume work for free. Given this, and in light of the numerous impossibility results that have long plagued approaches to achieve false-name-proofness in mechanism design [5], it might seem hopeless to design sybil-proof accounting schemes with transitive trust. However, experience from the design of reputation systems and credit networks has shown that, in certain settings and under certain assumptions, sybil-proof mechanisms do exist. Nevertheless, for accounting mechanisms with transitive trust, this has been an open question for a long time, with particular interest to the multi-agent systems community. For example, the designers of *Tribler* [14], a P2P file-sharing client that

Appears in: *Alessio Lomuscio, Paul Scerri, Ana Bazzan, and Michael Huhns (eds.), Proceedings of the 13th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2014), May 5-9, 2014, Paris, France.*

uses transitive trust, have already tried to design sybil-proof accounting mechanisms as early as 2007. Our paper now reveals the difficulty of this problem: simultaneously satisfying strong transitive trust and strong sybil-proofness is impossible, and we show where trade-offs must be made.

1.2 Problem Set-Up

We consider a distributed work system of n agents, each capable of doing work for each other. All agents provide the same quality of work, quantifiable in the same units. Performing work is costly and thus, all else equal, agents prefer to free ride rather than work. Work interactions are bilateral and private, i.e., no outside agent can observe or monitor an interaction. There are no binding contracts, and work and "payments" for work cannot be made simultaneously.

Periodically, each agent receives requests from some set of agents for a work contribution. Agents have no *a priori* preference in working for one agent over another. In particular, there are no pre-existing trust relationships between agents; trust links are only formed after one agent performs work for another. A direct trust link of weight w from A to B only means that in the future, A will give preferential treatment to B over other agents with scores less than w. If an agent has no trust links yet, it performs work for a random agent requesting work. An alternative way of bootstrapping the trust links is via simultaneous exchanges of small pieces of work (e.g., sharing file fragments in BitTorrent).

We assume a trusted center, and at any time, agents can make voluntary (and perhaps untruthful) reports to the center about work contributed and consumed in transactions involving themselves. Yet, there is no way to *bind* a report to the work contribution itself. At any time, an agent can query the center for all reports that have been made by all agents. Each agent has a unique identity, but for a tiny cost $\varepsilon > 0$, an agent can generate sybils.[1]

In our own prior work [21], we have formalized *accounting mechanisms*, which tally work performed and consumed, and compute a score that approximates an agent's net contributions. For this paper, we also use this accounting mechanism framework.

1.3 Overview of Results

The focus of this paper is a theoretical inquiry into the design of sybil-proof accounting mechanisms with transitive trust. We introduce a set of natural properties, and show which combinations of properties lead to impossibility results, and which combinations are possible. Our analysis reveals the necessary trade-off one must make between different notions of transitive trust and sybil-proofness in designing accounting mechanisms. Informally stated, and under some natural assumptions, our main results are as follows:

1. Every accounting mechanism that satisfies strong transitive trust is vulnerable to strongly beneficial sybil attacks.
2. If we relax this to a weaker form of transitive trust, then there exists an accounting mechanism that is robust to the strong kinds of sybil attacks.
3. However, as long as we require even a weak form of transitive trust, every accounting mechanism is still vulnerable to a weak kind of sybil attack.

By showing which properties of accounting mechanisms are inherently incompatible, we provide guidance for future research to focus on the optimal trade-off between different properties.

[1]In practice, sybils are generally costly to produce, for example because their creation involves solving CAPTCHAS to create new accounts, or because it requires the acquisition of multiple IP addresses. If the cost for creating sybils were zero, an agent could create an infinite number of sybils and exploit the bootstrapping process by consuming tiny pieces of work and then disappearing.

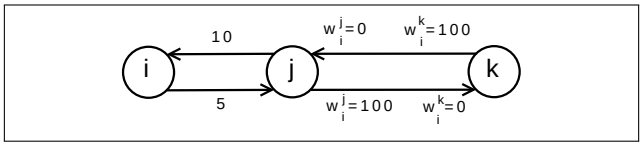

Figure 1: A subjective work graph from agent i's perspective. Edges where i has direct information only have one weight. Other edges can have two weights, corresponding to the possibly conflicting reports of the two agents involved.

2. FORMAL MODEL

The work performed by all agents in the distributed work system is captured by a work graph:

DEFINITION 1. ***(Work Graph)*** *A work graph $G = (V, E, w)$ has vertices $V = \{1, \ldots, n\}$, one for each agent, and directed edges $(i, j) \in E$, for $i, j \in V$, corresponding to work performed by i for j, with weight $w(i, j) \in \mathbb{R}_{\geq 0}$ denoting the number of units of work.*

We use $e \in E$ when referring to a generic edge, and $(i, j) \in E$ when referring to the specific edge from i to j. The true work graph may be unknown to the agents who only have direct information about their own participation:

DEFINITION 2. ***(Agent Information)*** *Each agent $i \in V$ keeps a private history $(w_i(i, j), w_i(j, i))$ of its interactions with other agents $j \in V$, where $w_i(i, j)$ and $w_i(j, i)$ are the work performed for j and received from j respectively.*

Based on its own experiences and reports from others, agent i can construct a subjective work graph (see Figure 1). Let $w_i^j(j, k), w_i^k(j, k) \in \mathbb{R}_{\geq 0}$ denote the edge weight, as reported by agent j and agent k respectively.

DEFINITION 3. ***(Subjective Work)*** *A subjective work graph from agent i's perspective, $G_i = (V_i, E_i, w_i)$, is a set of vertices $V_i \subseteq V$ and directed edges E_i. Each edge $(j, k) \in E_i$ for which $i \notin \{j, k\}$, is labeled with one, or both, of weights $w_i^j(j, k), w_i^k(j, k)$ as known to i. For edges (i, j) and (j, i) the associated weight is $w_i^i(i, j) = w(i, j)$ and $w_i^i(j, i) = w(j, i)$ respectively.*

We assume that the weights w are shared through voluntary reports to a central server, while still maintaining the core assumption of no central monitoring and no independent verification of reports. Thus, the edge weights $w_i^j(j, k)$ and $w_i^k(j, k)$ need not be truthful reports about $w(j, k)$. The assumption that the center cannot verify reports is motivated in two ways: first, some P2P systems (like Tribler) are serverless, such that "verification" is impossible. Second, even if a

server were used, it could receive conflicting reports from two agents, and we are not aware of any mechanism that could determine which agent is lying.

Periodically, an agent can receive a work request by a set of agents, which induces a choice set:

DEFINITION 4. ***(Choice Set)*** *We let $C_i \subseteq V \setminus \{i\}$ denote the choice set for agent i, i.e., the set of agents that are currently interested in receiving some work from i.*

An accounting mechanism computes a *score* for each j in choice set C_i, given the information contained in the subjective work graph.

DEFINITION 5. ***(Accounting Mechanism)*** *Accounting mechanism M takes as input a subjective work graph G_i, a choice set C_i, and determines the score $S_j^M(G_i, C_i) \in \mathbb{R}$, for any agent $j \in C_i$, as viewed by agent i.*

Throughout this paper we assume that the center runs the accounting mechanism and computes the scores. Nevertheless, the computation still uses each agent's *subjective* work graph to compute the corresponding scores.

We now introduce two natural assumptions that all accounting mechanisms must satisfy, and which will be essential for our impossibility results. First, we require that the scores do not depend on disconnected agents, i.e., adding or removing agents with no amount of work consumed or performed does not change the scores of other agents.

ASSUMPTION 1. ***(Independence of Disconnected Agents (IDA))*** *Accounting mechanism M satisfies* independence of disconnected agents *if, for any subjective work graph $G_i = (V_i, E_i, w_i)$ and any choice set C_i, for any $k \in V_i$ for which there does not exist an edge in E_i or for which all edges in E_i have zero weight, where G_i' denotes the graph where node k has been removed, the following holds:*

$$\forall j \in V_i' : S_j^M(G_i, C_i) = S_j^M(G_i', C_i').$$

The next assumption requires that a priori, the accounting mechanism does not put more or less trust into any agent, i.e., we only consider mechanisms that, for any renaming of the agents in the network, return the same scores.

ASSUMPTION 2. ***(Anonymity (ANON))*** *Accounting mechanism M satisfies* anonymity *if for any subjective work graph $G_i = (V_i, E_i, w_i)$ and choice set C_i, and graph isomorphism f such that $G_i' = f(G_i)$, $C_i' = f(C_i)$ and $f(i) = i$:*

$$\forall j \in V_i \setminus \{i\} : S_j^M(G_i, C_i) = S_{f(j)}^M(G_i', C_i').$$

Once the accounting mechanism has computed a score for each agent in the choice set, an agent uses its *allocation policy* to decide to whom to allocate work to (see Figure 2). Intuitively, we would expect that under any reasonable allocation policy the probability of an agent being allocated weakly increases if its score increases. Many different allocation policies are conceivable, including proportional allocation policies or threshold-based policies (see [19] for an experimental comparison of different allocation policies). To simplify the exposition, we will focus on the winner-take-all policy throughout this paper and also use it to prove our positive theoretical results.

Figure 2: Accounting Mechanism and Allocation Policy.

ASSUMPTION 3. ***(Winner-Take-All (WTA) Allocation Policy)*** *Given G_i, choice set C_i, and accounting scores $S_j^M(G_i, C_i)$ for each agent $j \in C_i$, the* winner-take-all allocation policy (WTA) *selects the agent with the highest score, i.e., $A(S^M(G_i, C_i)) = \arg\max_{k \in C_i} S_k^M(G_i, C_i)$, breaking ties at random, to receive one unit of work from i.*

Note that using the winner-take-all allocation policy, agents give preferential treatment to agents that have a high score. Thus, after a while, free-riders will not receive work anymore and have an incentive to perform work.

3. THE DROP-EDGE MECHANISM

We now present the first example of an actual accounting mechanism, adopted from [21]:

DEFINITION 6. ***(Drop-Edge Mechanism)*** *Given subjective work graph G_i and choice set C_i, construct the modified graph $G_i^D = (V_i, E_i, w_i^D)$ with the w_i^D defined as:*

$$\forall (j,k) | i \in \{j,k\} : w_i^D(j,k) = w_i^i(j,k)$$

$$\forall (j,k) | j,k \in C_i : w_i^D(j,k) = 0 \quad (1)$$

$$\forall (j,k) | j \in C_i, k \notin C_i : w_i^D(j,k) = w_i^k(j,k) \quad (2)$$

$$\forall (j,k) | k \in C_i, j \notin C_i : w_i^D(j,k) = w_i^j(j,k) \quad (3)$$

$$\forall (j,k) | j,k \notin C_i, i \notin \{j,k\} : w_i^D(j,k) = \max\{w_i^j(j,k), w_i^k(j,k)\}.$$

Missing reports in the max-operator are set to 0. Agent j's score is $S_j^D(G_i, C_i) = MF_{G_i^D}(j,i) - MF_{G_i^D}(i,j)$, where $MF_{G_i^D}(j,i)$ denotes the maximum flow from j to i in G_i^D.

Lines (1)-(3) implement the simple "edge-dropping" idea. Any reports received by agent i from agents in the choice set C_i are dropped in determining edge weights in modified graph G_i^D. An edge (j,k) is dropped completely if both j and k are inside C_i. See Figure 3 for an illustration.

It is easy to verify that Drop-Edge satisfies IDA (because max-flow is not affected when disconnected agents are removed) as well as ANON (because max-flow is anonymous).

PROPOSITION 1. *The Drop-Edge mechanism satisfies IDA and ANON.*

3.1 Transitive Trust

Now we want to know what notion of *transitive trust* the Drop-Edge mechanism provides. It turns out that Drop-Edge satisfies a very strong notion of transitive trust: once you trust agent j, you also trust all agents you are "referred" to by j.

DEFINITION 7. ***(Strong Transitive Trust)*** *Accounting mechanism M satisfies* strong transitive trust *if, for every subjective work graph $G_i = (V_i, E_i, w_i)$, there exists a $j \in V_i$, an amount of work W_j and an amount of work W_k, such that for any set of new nodes $K = \{k_1, ..., k_n\}$ added to G_i such that $G_i' = (V_i', E_i, w_i)$ with $V_i' = V_i \cup K$, and*

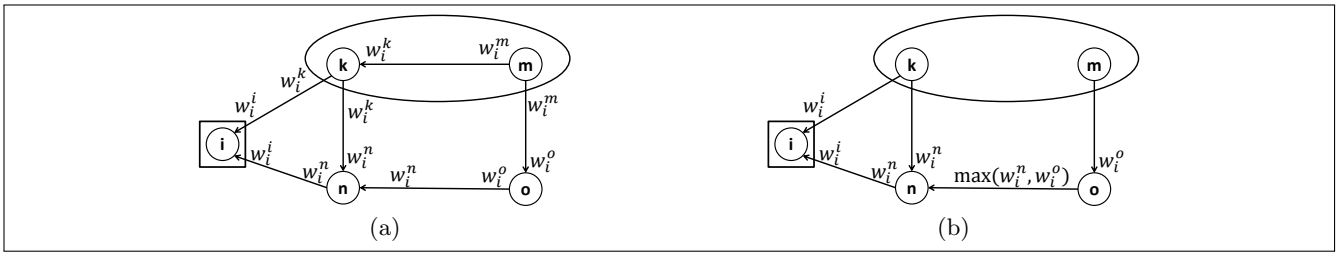

Figure 3: An illustration of the Drop-Edge mechanism from i's perspective. The choice set is $C_i = \{k, m\}$. (a) Agent i's subjective work graph where each edge has two weights, one from each agent who knows about that edge. (b) Agent i's subjective work graph after the Drop-Edge mechanism has been applied.

- *if j performs W_j units of work for i, and*
- *each $k \in K$ performs W_k units of work for j, and*
- *j makes a truthful report about the work received from each $k \in K$, leading to G_i'',*

then for every choice set C_{k_l} that contains k_l but neither j nor any of the other agents from K, it holds that: $A(S^M(G_i'', C_{k_1})) = k_1$ and after k_1 consumes 1 unit of work from i leading to G_i''' it holds that $A(S^M(G_i''', C_{k_2})) = k_2$, etc., and after k_{n-1} consumes 1 unit of work from i leading to G_i^{n+1}, it holds that $A(S^M(G_i^{n+1}, C_{k_n})) = k_n$.

PROPOSITION 2. *The Drop-Edge mechanism together with WTA satisfies strong transitive trust.*

PROOF. Given graph $G_i = (V_i, E_i, w_i)$, take any $j \in V_i$ and let $W_j = W_k = S_{\max} + 1$ where $S_{\max}$ is the maximum score any $l \in V_i$ could get for any choice set C_i. We add a set of agents K to G_i, which does not change the scores (IDA). Now j performs W_j units of work for i and each $k \in K$ performs W_k units of work for j, and j reports this truthfully, leading to G_i'. Using Drop-Edge, the scores for each $k \in K$ given choice set C_k are: $S_k^D(G_i', C_k) = MF_{G_i'^D}(k, i) - MF_{G_i'^D}(i, k) = S_{\max} + 1$. Thus, given choice set C_{k_1} and using the WTA allocation policy, agent k_1 receives 1 unit of work from i. This changes the flow between k_1 and i, but does not affect any other agent in K. Thus, we can continue this process for all n agents, each receiving 1 unit of work. □

3.2 Misreport-Proofness

Ideally, accounting mechanisms should be robust against strategic manipulations. Otherwise, agents may try to manipulate the mechanism to gain an advantage, which can lead to suboptimal allocations of resources and reduce overall efficiency. The first class of manipulations we consider are *misreports*, where an agent reports false information about its work performed or consumed. In words, a mechanism is misreport-proof if reporting false work information can only worsen an agent's own score or improve the score of other agents.

DEFINITION 8. ***(Long-term Misreport-proof)***[2] *An accounting mechanism M satisfies long-term misreport-proofness if, for any subjective work graph G_i, any choice set C_i, any agent $j \in C_i$, for every misreport manipulation m_j by j, and any set of interactions I by all agents that comes after the misreport manipulation, such that $G_i'' = G_i \downarrow m_j \downarrow I$, and $G_i' = G_i \downarrow I$, the following holds:*

- $S_j^M(G_i'', C_i) \leq S_j^M(G_i', C_i)$, *and*
- $S_k^M(G_i'', C_i) \geq S_k^M(G_i', C_i)\ \forall k \in C_i \setminus \{j\}$.

Note that the definition of long-term misreport-proofness does not rule out that the choice set changes over time. We only compare the scores (and use the choice set) at one point in time, namely after the manipulation and all interactions are over.

PROPOSITION 3. *The Drop-Edge mechanism is long-term misreport-proof.*

PROOF. In [21], we have already shown that using Drop-Edge, j's reports have no direct impact on its own or other agents' scores from i's perspective whenever j is in i's choice set. Furthermore, j only has an indirect impact on another agent k in situations where k is inside someone's choice set and j is not. In that situation, j's report may make a difference in the allocation decision. However, given the same set of interactions I, neither its own nor other agents' scores change when it would matter for j, which shows long-term misreport-proofness. □

3.3 Sybil Attacks

The second class of manipulations we consider are *sybil manipulations*, where an agent introduces sybils (fake agents) into the network to manipulate the accounting mechanism. Given subjective work graph G_i, an attacking agent can do multiple things, e.g., add sybils to the network, or make multiple false reports. We model this as happening in one step, inducing a new graph G_i'. Note that it is irrelevant how an attacker creates sybils. In our model and for our analysis, the only thing that matters is how the attack affects the subjective work graph.

DEFINITION 9. ***(Sybil Attack)*** *A sybil attack by agent j is a tuple $\sigma_j = (V_s, E_s, w_s)$ where $V_s = \{s_{j_1}, s_{j_2}, ...\}$ is a set of sybils, $E_s = \{(x, y) : x, y \in S \cup \{j\}\}$, and w_s are the edge weights for the edges in E_s (one weight per edge). Applying the sybil attack σ_j to agent i's subjective work graph $G_i = (V_i, E_i, w_i)$ results in a modified graph $G_i \downarrow \sigma_j = G_i' = (V_i \cup V_s, E_i \cup E_s, w')$ where $w'(e) = w_i(e)$ for $e \in E_i$ and $w'(e) = w_s(e)$ for $e \in E_s$.*

We now present an example that illustrates how vulnerable the Drop-Edge mechanism is to sybil attacks, before we turn to the theoretical analysis in the next section.

[2]Note that we extend the definition of *misreport-proofness* from [21] to *long-term misreport-proofness* because sybil attacks require a more dynamic analysis.

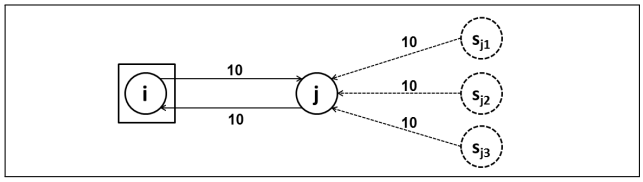

Figure 4: A sybil attack on Drop-Edge.

EXAMPLE 1. *Consider Figure 4 where we present a sybil attack on the Drop-Edge mechanism. Agent j has already performed/consumed 10 units of work for/from agent i, and we assume that i now trusts j's reports about other agents to some degree. Now, j creates 3 sybils and falsely reports to i that these sybils have performed 10 units of work for j. Assuming that this raises the sybils' scores high enough, the sybils can exploit their scores and consume work from i.*

4. THEORETICAL ANALYSIS

In this section, we analyze whether other accounting mechanisms are more robust against sybil attacks than Drop-Edge or whether this vulnerability is unavoidable. For this analysis, we distinguish between weakly beneficial sybil attacks and strongly beneficial attacks. Note that the following definitions are purposefully written to be agnostic to the behavior of other agents in the network. Our theoretical results hold without requiring specific assumptions about their behaviors. We consider a sybil attack to be (weakly) *beneficial* if as a result of the manipulation, the attacking agent or one of its sybils receives some work when it previously didn't, without a negative effect on the attacking agent:

DEFINITION 10. ***((Weakly) Beneficial Sybil Attack)*** *Given accounting mechanism M, subjective work graph G_i, and choice set C_i such that $A(S^M(G_i, C_i)) = k$, a (weakly) beneficial sybil attack σ_j by agent $j \neq k \in V_i$ such that $G_i \downarrow \sigma_j = G'_i$ and $C_i \downarrow \sigma_j = C'_i$, is one where at least one of (1), (2), or (3), or combinations thereof holds:*

(1) j's score is increased so that $A(S^M(G'_i, C'_i)) = j$.

(2) other agents' scores are lowered so that $A(S^M(G'_i, C'_i)) = j$.

(3) sybil s is created with a score so that $A(S^M(G'_i, C'_i)) = s$, whereby j's score is not decreased.

Note that our definition of a sybil attack assumes that the sybil attack itself does not involve performing any work for other agents; the sybil attack only creates sybils and fake edges. However, it may be necessary for an attacker to first perform some work, to bring itself into a position where a sybil attack is beneficial. How beneficial an attack really is depends on the trade-off between the amount of work performed and the amount of work consumed:

DEFINITION 11. ***(Strongly Beneficial Sybil Attacks)*** *Given accounting mechanism M and work graph $G_i = (V_i, E_i, w_i)$, assume agent $j \in V_i$ performs ω_j units of work for agents in V_i to increase its score, leading to G'_i. Then j performs a sybil attack σ_j such that $G''_i = G'_i \downarrow \sigma_j$. Let $(\omega_j, \sigma_j)^n$ denote an $n-$times-repetition of this process, i.e., of performing some work and performing the sybil attack. Let $\omega^n_-(\sigma_j) > 0$ denote the amount of work involved in performing the n-times repetition of this process, and let $\omega^n_+(\sigma_j)$ denote the amount of work that agent j or any of its sybils will be able to consume as a result of the process. We call σ_j a strongly beneficial sybil attack if:* $\lim_{n\to\infty} \frac{\omega^n_+(\sigma_j}{\omega^n_-(\sigma_j} = \infty$.

Note that the sybil attack presented in Figure 4 was actually strongly beneficial. Agent j could continue generating new sybils that can then consume work for free. We will now show that this is not unique to Drop-Edge, but that in fact all mechanisms that are long-term misreport-proof and satisfy strong transitive trust are this vulnerable to sybil attacks. The intuition for this is simple: because of strong transitive trust, an agent can create an infinite number of sybils that inherit some of the trust that other against have placed in j, and because of the misreport-proofness property, agent j cannot be penalized for making false reports about its sybils.

THEOREM 1. *Every accounting mechanism that satisfies IDA, ANON, long-term misreport-proofness, and strong transitive trust, is vulnerable to strongly beneficial sybil attacks.*

PROOF. Assume that accounting mechanism M satisfies the strong transitive trust property, and take any subjective work graph G_i with nodes j and $K = \{k_1, \ldots, k_n\}$ as described in Definition 7. Let W_j and W_k be the corresponding values, such that j performs W_j units of work for i, and all agents $k \in K$ perform W_k units of work for j. Because M is long-term misreport-proof, j is best off making a truthful report regarding the work received, leading to G'_i such that now $A(S^M(G'_i, C_{k_1})) = k_1$ etc., as described in Definition 7. Now, assume that agent j creates n sybils $s_1, \ldots, s_n$, which are added to the graph, leading to G''_i. Because of the independence of disconnected agents (Assumption 1), this does not change any of the scores and thus, the result of the allocation policy also does not change. Because M is anonymous (Assumption 2), we can apply a graph isomorphism f to G''_i that only switches the labels of all nodes $k_1, \ldots, k_n$ with $s_1, \ldots, s_n$, leading to G'''_i and C'_{s_1}. Thus, now $A(S^M(G'''_i, C'_{s_1})) = s_1$, and after sybil s_1 consumes 1 unit of work from i, it holds that $A(S^M(G''''_i, C''_{s_2})) = s_2$, and so on. Thus, property (3) of Definition 10 is satisfied. Of course, j can also directly perform a sybil attack σ_j, adding n agents to the graph and reporting that all of them have performed W_k units of work for j, with the same outcome. Note that the work to perform this sybil attack is fixed at $\omega^n_-(\sigma_j) = W_j$, which, in particular, is independent of n. As we increase the number of sybils n, each additional sybil receives at least one unit of work from i. Thus, $\lim_{n\to\infty} \frac{\omega^n_+(\sigma_j)}{\omega^n_-(\sigma_j)} = \infty$. □

We see that the strong transitive trust property asks too much. For this reason we introduce *weak transitive trust*, where now agent j can only transfer its trust from i to *one* other agent k, instead of to an infinite number of agents:

DEFINITION 12. ***(Weak Transitive Trust)*** *Accounting mechanism M satisfies weak transitive trust if, for every subjective work graph $G_i = (V_i, E_i, w_i)$, there exists a $j \in V_i$, after adding node k to G_i this leads to $G'_i = (V'_i, E_i, w_i)$ with $V'_i = V_i \cup \{k\}$, and there exists an amount of work W_j and W_k, such that if j performs W_j units of work for i, and k performs W_k units of work for j, and j makes a truthful report about the work received from k, leading to G''_i, then for every choice set C_k that contains k but not j, it holds that: $A(S^M(G''_i, C_k)) = k$.*

We now show that we have actually gained something by relaxing *strong* transitive trust to *weak* transitive trust: we can now construct accounting mechanisms that are robust against strongly beneficial sybil attacks.

DEFINITION 13 (DROP-EDGE-VARIANT-1). *Given subjective work graph $G_i = (V_i, E_i, w_i)$ and choice set C_i, construct the modified graph G_i^D with weights w_i^D as defined for Drop-Edge. Additionally, we temporarily create* trust scores *$t_i(j,k)$ for each edge (j,k), which are initialized to zero, and then updated in the following way:*

1. *For each unit of work received from agent j: $t_i(j,i)$++;*
2. *For each unit of work performed for agent j by i:*
   (a) *Compute the max-flow from j to i based on G_i^D.*
   (b) *For each of i's incoming edges (k,i) that are part of this max-flow with $flow(k,i) > 0$: $t_i(k,i)$- -;*
3. *To compute agent j's score, temporarily construct a new graph G_i^{D1} based on G_i^D with:*
   (a) *for all edges (k,i) with $k \neq j : w_i^{D1}(k,i) = t_i(k,i)$*
   (b) *for all other edges $(x,y) : w_i^{D1}(x,y) = w_i^D(x,y)$.*

j's score is $S_j^{D1}(G_i^{D1}, C_i) = MF_{G_i^{D1}}(j,i) - MF_{G_i^{D1}}(i,j)$.

THEOREM 2. *The Drop-Edge-Variant-1 accounting mechanism together with allocation policy WTA satisfies IDA, ANON, long-term misreport-proofness, and weak transitive trust, and is robust against strongly-beneficial sybil attacks.*

PROOF. First, the mechanism satisfies IDA because the max-flow between two nodes is not affected when disconnected agents are removed from the graph. Second, the mechanism satisfies ANON because max-flow is anonymous. Third, the mechanism inherits long-term misreport-proofness from Drop-Edge. When i computes the score for j, then the additional trust-score operations and the construction of G_i^{D1} only affect the underlying work weights on edges (k,i) with $k \neq j$. Thus, no report by j is ever used to change the underlying work weights when it matters for j, which is needed for long-term misreport-proofness.

Fourth, the mechanism satisfies the weak transitive trust property, because Drop-Edge satisfies strong transitive trust (see Proposition 2) and in Drop-Edge-Variant-1, the trust scores are initialized to be 0. In particular, a completely new node j that has never reported anything is unaffected by the additional trust score computations. Thus, j can perform enough units of work for i to gain i's trust and then make a positive report about some new node k to i, such that k will at least receive one unit of work from i.

Fifth, the mechanism is robust against strongly beneficial sybil attacks because of its use of the trust scores. The trust scores $t_i(j,i)$ are used to make sure that i does not perform too many units of work for other agents k because of positive reports from j about k.[3] Every time one unit of work is performed (or received), the trust score $t_i(j,i)$ is increased or decreased appropriately. Thus, no matter how much work W_j agent j initially performed for i, if j tries to exploit i via a sybil attack, then at some point i's trust in j will be "used up," and from then on, agent i will not perform any more work for other agents k based on reports by j, which rules out strongly beneficial sybil attacks. □

[3]We know that our algorithm decreases the trust scores of multiple edges in one round, while it could also compute the relative weight of the edges in the max-flow and decrease the trust scores accordingly (more slowly). However, this aspect is not important for the theoretical result. For practical considerations and a more sophisticated algorithm see [7].

Unfortunately, the relaxation from *strong* to *weak transitive trust* does not rule out all kinds of sybil attacks:

THEOREM 3. *Every accounting mechanism that satisfies IDA, ANON, long-term misreport-proofness, and weak transitive trust, is vulnerable to weakly beneficial sybil attacks.*

PROOF. The proof proceeds similarly as the proof of Theorem 1. We consider any accounting mechanism M that satisfies weak transitive trust, and take any subjective work graph G_i with nodes j and k as described in Definition 12. Let W_j and W_k be the corresponding values, such that j performs W_j units of work for i, and k perform W_k units of work for j. Because M is long-term misreport-proof, j is best off making a truthful report regarding the work received, leading to G_i' such that now $A(S^M(G_i', C_k)) = k$, as described in Definition 12. Now, assume that agent j creates a sybil agent s which is added to the graph, leading to G_i''. Because of the independence of disconnected agents (Assumption 1), this does not change any of the scores and thus, the result of the allocation policy also does not change. Because M is anonymous (Assumption 2), we can apply a graph isomorphism f to G_i'' that only switches the labels of node k with s leading to G_i''' and C_s'. Thus, now $A(S^M(G_i''', C_s')) = s$, i.e., sybil s can consumes 1 unit of work from i, and because of long-term misreport-proofness, this does not have a negative effect on j. Thus, property (3) of Definition 10 is satisfied. Of course, j can also directly perform a sybil attack σ_j, adding sybil agent s to the graph and reporting that s has performed W_k units of work for j, with the same outcome. Thus, this constitutes a weakly beneficial sybil attack. □

5. TIGHTNESS

What Theorem 3 tells us is that given the two assumptions IDA and ANON, it is impossible to simultaneously satisfy the three properties (1) long-term misreport-proofness, (2) weak transitive trust, and (3) robustness against weakly beneficial sybil attacks. In this section, we show that Theorem 3 is tight, in the sense that dropping any one of these three properties leads to a positive result.

PROPOSITION 4. *There exists an accounting mechanism that satisfies IDA, ANON, long-term misreport-proofness, and weak transitive trust.*

PROOF. Consider the basic Drop-Edge mechanism presented in Section 3 together with WTA. By Proposition 1, it satisfies IDA and ANON. By Proposition 2, it also satisfies the strong transitive trust property which implies that it satisfies weak transitive trust. Finally, by Proposition 3, it is also long-term misreport-proof. Note, however, that it is vulnerable to strongly beneficial sybil attacks (Example 1), which is unavoidable by Theorem 1. □

PROPOSITION 5. *There exists a mechanism that satisfies IDA, ANON, and weak transitive trust, and is robust against weakly beneficial sybil attacks.*

PROOF. We create a new mechanism, which we call *Drop-Edge-Variant-2*, which works like the basic Drop-Edge mechanism as presented in Section 3, except:

- After any agent k consumes one unit of work from i, we compute the max-flow from k to i, and for each edge $e = (j,i)$ that is part of this max-flow we set: $w_i^{D2}(j,i) = w_i^{D2}(j,i) - 1$.
- j's score is $S_j^{D2}(G_i^{D2}, C_i) = MF_{G_i^{D2}}(j,i) - MF_{G_i^{D2}}(i,j)$.

Thus, Drop-Edge-Variant-2 is even more resolute than Drop-Edge-Variant-1 in how it handles indirect effects: it reduces the weight $w_i^{D2}(j,i)$ on edge (j,i), even if j has not consumed any new work from i, but just because j was indirectly responsible for another agent k receiving work from i, due to a positive report by j about k.

First, Drop-Edge-Variant-2 together with WTA satisfies IDA (because removing disconnected agents has no influence on the max-flow between two nodes), and ANON (because max-flow is anonymous).

Second, to show weak transitive trust, consider any graph $G_i = (V_i, E_i, w_i)$ and any $j \in V_i$, and let $W_j = S_{\max} + 1$, where $S_{\max}$ is the maximum score any $l \in V_i$ could get for any choice set C_i. Now, let j perform W_j units of work for i, holding everything else constant (i.e., no other agent including j is consuming anything). This increases j's score by $S_{\max} + 1$. Now, consider a new agent k which performs $W_k = W_j$ units of work for j. If j reports this truthfully to agent i, then k's score from i's perspective is now at least $S_{\max} + 1$. Thus, given allocation policy WTA, agent k can now consume at least one unit of work from i, which shows that the weak transitive trust property is satisfied.

Finally, consider agent j trying to perform a sybil attack. By creating new sybil nodes with edges between those sybils and j, none of the max-flows between existing agents and agent i are affected, and thus conditions (1) and (2) of Definition 10 do not apply. But most importantly, if agent j now makes a positive report to agent i about work received from one of its sybils s, and if as a consequence $A(S^{D2}(G_i', C_i')) = s$, then sybil node s may receive one unit of work from i, but in turn, agent j's score is also lowered by one unit, according to the definition of Drop-Edge-Variant-2. Thus, condition (3) of Definition 10 is also not satisfied, which shows that the mechanism is robust against weakly beneficial sybil attacks. However, note that this comes at the cost of not being long-term misreport-proof. □

PROPOSITION 6. *There exists an accounting mechanism that satisfies IDA, ANON, long-term misreport-proofness, and is robust against weakly beneficial sybil attacks.*

PROOF. Consider an accounting mechanism that always returns the same score (e.g., 0) for all agents. Obviously, this mechanism satisfies IDA and ANON, is long-term misreport-proof as well as robust against weakly beneficial sybil attacks, because the agents' reports have no influence on the scores. However, note that the mechanism does not even satisfy the weak transitive trust property. □

6. DISCUSSION AND RELATED WORK

In this section, we contrast our results with other approaches and survey some related work. First, if binding contracts or atomic transactions were available, then real currencies, micro payment systems, or electronic currencies like BitCoin [16] could be used. These would be sybil-proof because I would need to transfer some currency to my sybils to enable them to consume work, which is not advantageous. Locally-valid currencies like iOwe [11] could be implemented in our domain. However, while iOwe is sybil-proof, it does not provide transitive trust.

An emerging literature on *credit networks* (e.g.,[15, 22, 13, 6]) considers similar challenges as we do. However, these papers assume pre-existing trust networks, and they also do not have a real notion of *transitive* trust. In our model, after agent A has performed work for B, agent B is willing to report this to the system given a misreport-proof mechanism. However, agent B is not personally vouching for A, or taking a risk on behalf of A. In contrast, in credit networks, a trust link between A and B means that agent B is willing to pay agent C (in IOUs) for work that A consumes from C, trusting that A will repay B in the future. This explains why credit networks are sybil-proof. If I created a sybil and vouched for it, then I would have to pay for the work that my sybil consumes, which is not beneficial. This shows that credit networks are an interesting way to leverage an underlying social network. Similarly, *SybilGuard* [23] also leverages an underlying social network, but not as a basis for a credit network, but instead to automatically bound the number of effective sybil identities that can be created.

The work on transitive trust and reputation mechanisms [10, 1] is an important precursor to our work. Cheng et al. [2, 3] have studied the sybil-proofness of reputation mechanisms. While this work influenced our thinking about sybil-proofness, unfortunately, their results do not translate to our domain. In distributed work systems, every *positive* report by A about his interaction with B, i.e., B performed work for A, is simultaneously a *negative* report about A, i.e., A received work from B. This fundamental tension is not present in reputation mechanisms. Resnick and Sami [18] also study the sybil-proofness problem. However, in their model, when two agents interact, the principal agent observes whether the outcome (due to the other agent's action) is positive or negative. Thus, a defector can immediately be identified, in contrast to our setting.

One of the largest steps forward regarding robust incentives in real-world multi-agent systems was the BitTorrent protocol [4]. However, BitTorrent only promotes bilateral transactions and does not satisfy transitive trust. Feldman et al. [8, 9] identify numerous challenges involved in providing robust incentives in fully decentralized P2P networks. Piatek et al. [17] find empirically that most users of P2P file-sharing networks are connected via a one hop link and motivate the use of well-connected intermediaries to broker information, enabling transitivity. Along similar lines, Meulpolder et al. [14] present a fully decentralized mechanism, but without a formal analysis of its properties.

Note that none of the prior literature on sybil attacks, including a survey by Levine et al. [12] on "Solutions to the Sybil Attack," had previously made the connection between sybil-proofness and transitive trust. Thus, our work provides new insights regarding the difficulties involved in achieving sybil-proofness that were previously not known.

7. CONCLUSION

In this paper, we have studied the design of sybil-proof accounting mechanisms with transitive trust. Our domain is a distributed work system in which all transactions are private and bilateral, transactions cannot be bound to payments or reports, and there are no pre-existing trust links. The mechanisms we study can be used in the context of many multi-agent-system applications. For example, our approach is applicable to P2P file-sharing networks, ride-sharing systems, and ad-hoc wireless routing networks.

Our main results illustrate an interesting trade-off between different notions of transitive trust on the one side and different levels of robustness against sybil attacks on

the other side. We have shown that it is impossible to simultaneously require the strongest form of transitive trust while also requiring robustness against strongly beneficial sybil attacks. However, we have also demonstrated that it is possible to strike a balance, by designing mechanisms that provide a weaker form of transitive trust while being robust against strongly beneficial sybil attacks.

The particular mechanism we have presented only serves as a proof-of-concept, but further refinements are necessary to adapt the mechanism for use in practice. For future research, we are considering an experimental evaluation of this approach, comparing the effects of strong vs. weak transitive trust on overall efficiency. Delaviz et al. [7] have independently developed a mechanism that is similar to ours, and they have already provided some experimental results suggesting that the general approach may be useful in practice.

Our impossibility results also illustrate the usefulness of social ties for providing robustness against sybils. In domains with existing trust relationships, e.g., based on social networks, a different form of transitive trust can be enabled, while providing robustness to sybil attacks. Liu et al. [13] have shown, that even using just a limited amount of direct trust, high efficiency gains can be realized. However, in many domains it is unrealistic to assume pre-existing trust relations, which limits the applicability of such approaches. We have shown what notions of sybil-proofness can be achieved, even in domains where no pre-existing trust relations exist. This significantly broadens the applicability of our mechanisms compared to mechanisms that require pre-existing trust. By showing which properties of accounting mechanisms are inherently incompatible, future research can focus on the optimal trade-off between those properties.

Acknowledgments

We thank Michel Meulpolder, Johan Pouwelse, Ian Kash, Ariel Procaccia, and Mike Ruberry for helpful discussions. An earlier version of this paper was presented at NetEcon'11, but without archival proceedings [20]. We would like to thank the participants of this workshop for helpful questions and comments. Furthermore, we are thankful to the anonymous reviewers for their helpful feedback. This work was supported in part by NSF grant CCF-0915016.

8. REFERENCES

[1] A. Altman and M. Tennenholtz. An Axiomatic Approach to Personalized Ranking Systems. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

[2] A. Cheng and E. Friedman. Sybilproof Reputation Mechanisms. In *Proceedings of the ACM SIGCOMM Workshop on Economics of Peer-to-Peer Systems (P2PECON)*, August 2005.

[3] A. Cheng and E. Friedman. Manipulability of PageRank under Sybil Strategies. In *Proceedings of the 1st Workshop of Networked Systems (NetEcon06)*, June 2006.

[4] B. Cohen. Incentives Build Robustness in BitTorrent. In *Proceedings of the Workshop on Economics of Peer-to-Peer Systems (P2PEcon)*, Berkeley, CA, June 2003.

[5] V. Conitzer and M. Yokoo. Using Mechanism Design to Prevent False-Name Manipulations. *AI Magazine, Special Issue on Algorithmic Game Theory*, 31(4):65–77, 2010.

[6] P. Dandekar, A. Goel, R. Govindan, and I. Post. Liquidity in Credit Networks: A Little Trust Goes a Long Way. In *Proceedings of the 12th ACM Conference on Electronic Commerce (EC)*, San Jose, CA, June 2011.

[7] R. Delaviz, N. Andrade, J. A. Pouwelse, and D. H. Epema. SybilRes: A sybil-resilient flow-based decentralized reputation mechanism. In *Proceddings of the 32nd IEEE International Conference on Distributed Computing Systems*, 2012.

[8] M. Feldman, K. Lai, I. Stoica, and J. Chuang. Robust Incentive Techniques for Peer-to-Peer Networks. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, New York, NY, May 2004.

[9] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica. Free-Riding and Whitewashing in Peer-to-Peer Systems. *IEEE Journal on Selected Areas in Communications*, 24(5):1010–1019, 2006.

[10] E. Friedman, P. Resnick, and R. Sami. Manipulation-Resistant Reputation Systems. In N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, editors, *Algorithmic Game Theory*, pages 677–698. Cambridge University Press, New York, NY, 2007.

[11] D. Levin, A. Schulman, K. Lacurts, N. Spring, and B. Bhattacharjee. Making Currency Inexpensive with iOwe. In *Proceedings of the Workshop on the Economics of Networks, Systems, and Computation (NetEcon)*, 2011.

[12] B. N. Levine and C. S. ad N. Boris Margolin. A Survey of Solutions to the Sybil Attack. Technical Report.

[13] Z. Liu, H. Hu, Y. Liu, K. Ross, Y. Wang, and M. Mobius. P2P Trading in Social Networks: The Value of Staying Connected. In *Proceedings of IEEE Conference on Computer and Communications (INFOCOM)*, 2010.

[14] M. Meulpolder, J. Pouwelse, D. H. Epema, and H. J. Sips. BarterCast: A Practical Approach to Prevent Lazy Freeriding in P2P Networks. In *Proceedings of the 6th International Workshop on Hot Topics in Peer-to-Peer Systems (Hot-P2P)*, Rome, Italy, May 2009.

[15] A. Mislove, A. Post, K. P. Gummadi, and P. Druschel. Ostra: Leverging trust to thwart unwanted communication. In *Proceedings of the 5th Symposium on Networked Systems Design and Implementation (NSDI'08)*, San Francisco, CA, April 2008.

[16] S. Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash System. Available online at: bitcoin.org/bitcoin.pdf., 2008.

[17] M. Piatek, T. Isdal, A. Krishnamurthy, and T. Anderson. One Hop Reputations for Peer to Peer File Sharing Workloads. In *Proceedings of the 5th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 1–14, San Francisco, California, April 2008.

[18] P. Resnick and R. Sami. Sybilproof Transitive Trust Protocols. In *Proceedings of the 10th ACM Conference on Electronic Commerce (EC)*, Stanford, CA, July 2009.

[19] S. Seuken, M. Meulpolder, D. C. Parkes, J. A. Pouwelse, J. Tang, and D. H. J. Epema. Work Accounting Mechanisms: Theory and Practice. Working Paper. Department of Informatics, University of Zurich, 2014.

[20] S. Seuken and D. C. Parkes. On the Sybil-Proofness of Accounting Mechanisms. In *Proceedings of the Workshop on the Economics of Networks, Systems and Computation (NetEcon)*, San Jose, CA, June 2011.

[21] S. Seuken, J. Tang, and D. C. Parkes. Accounting Mechanisms for Distributed Work Systems. In *Proceedings of the 24th Conference on Artificial Intelligence (AAAI)*, Atlanta, GA, July 2010.

[22] N. Tran, B. Min, J. Li, and L. Subramanian. Sybil-Resilient Online Content Voting. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2009.

[23] H. Yu, M. Kaminsky, P. B. Gibbons, and A. Flaxman. SybilGuard: Defending Against Sybil Attacks via Social Networks. In *Proceedings of the ACM Conference On Computer Communications (SIGCOMM)*, 2006.