On the Sybil-Proofness of Accounting Mechanisms in P2P Networks

Alexander Stannat

On the Sybil-Proofness of Accounting Mechanisms in P2P Networks

by

Alexander Stannat

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Tuesday April 6, 2020 at 13:30 AM.

Student number:	4748239	
Project duration:	April 1, 2019 – April 6, 2020	
Thesis committee:	Dr. J. Pouwelse,	TU Delft, supervisor
	Dr. D. C. Gijswijt,	TU Delft
	Prof. Dr. D.H.J Epema,	TU Delft
	Dr. R.J. Fokkink,	TU Delft

This thesis is confidential and cannot be made public until April 06, 2020.

Abstract

Online P2P file sharing networks rely on the cooperation of participants to function effectively. Agents up- and download files to one another without the need for any central authority. If agents all contribute to the network and share roughly the same amounts of data as they contribute the network will operate, however if some agents decide to defect and consume far more resources than they contribute the file sharing will stagnate. In online networks with some kind of central authority, such as Ebay, Airbnb, etc. cooperation is achieved through a review system, which is maintained and secured by the central authority. P2P networks are however distributed and cooperation must be achieved without this central mitigator. One way of approaching this problem is by observing cooperative biological communities in nature. One finds that cooperation among biological organisms is achieved through a mechanism called indirect reciprocity. Indirect reciprocity is based on a reputation scheme in which agents share information about each other's cooperativeness aiding one another in deciding who to interact with and who to shun. In this work we analyse properties a reputation mechanism must satisfy in order to achieve cooperation in P2P networks, incentivising contributions and penalising excessive comsumption of data. In particular, we determine under what conditions reputation mechanisms are resistant to attacks on the P2P network. We focus on one attack above all, namely that of a sybil attack in which a malicious agent creates multiple fake identities who report high levels of cooperativeness about one another. We determine properties accounting mechanisms must satisfy in order to prevent attackers from obtaining arbitrarily high reputation and to consequently be able to consume arbitrarily large amounts of data. This thesis offers a theoretical framework for evaluating the effectiveness of reputation mechanisms on the basis of their ability to induce cooperation and their resistance to sybil attacks.

Preface

This thesis project is the culmination of master's degree in applied mathematics at the TU Delft. During my time at the TU's distributed systems group I was able to contribute to the open source and anonymous P2P file sharing network *Tribler*, both as a research assistant as well as with the contents of this master's thesis. I am very happy to have been able to apply my mathematical knowledge to a real-life setting in the field of computer science and hope that my contributions made to this field will prove useful to further research conducted at this group. In particular, I'm excited about the possibility of my research advancing the *Tribler* P2P file sharing platform which beyond simply facilitating the distribution of digital media, has made big steps in ensuring freedom of communication and information transfer as well as preserving the autonomy of peoples' digital lives under oppressive and authoritarian regimes. As a firm believer in the openness and freedom of the Internet I could not speak more highly of the values and endeavours of the Tribler development team and its head Johan Pouwelse, who along with Dion Gijswijt, has been my daily supervisor in this project.

I would like to express my deepest gratitude to both of my daily supervisors for their guidance and input throughout this thesis project. Johan Pouwelse for his infectious excitement and motivation delivered on a daily basis as well as his input throughout weekly meetings and brainstorming sessions in which he provided me with many thought-provoking impulses. Dion Gijswijt for the many fruitful meetings in which he helped me futher my work with high precision and sharpness. I would also like to thank Johan for facilitating a research visit at the Sokendai graduate school of advanced studies in Japan, in which I was able to collaborate with Hisashi Ohtsuki and his team of evolutionary biologists and learn about reputation mechanisms in networks of biological organisms. While this research was highly interesting, it unfortunately did not fit into the narrative of this thesis project and is therefore listed in the appendix. I would also like to thank members of the distributed systems group Martijn de Vos, Quinten Stokkink, Can Umut Ileri and Bulat Nasrulin for proofreading my work and their important input during brainstorming sessions. The team created a very pleasant and supportive working environment. The many (many) coffee breaks were always an enjoyable time and the smell of coffee will forever trigger memories of Quinten's charming sarcasm in my mind. I'd also like to thank my two friends Jetse Brouwer and Sacheendra Talluri for the fun times together on the third floor and at the "/pub". Lastly, I'd like to thank my family. Their continuous motivation and support has encouraged me to work hard and push through times of discouragement.

Alexander Stannat
Den Haag, April 2020

Contents

1 Introduction

Honest cooperation in a population is a requirement for any level of organisation to be reliably reached. From genes, unicellular organisms, and multicellular organisms to insect colonies and human societies, the ability to cooperate is of vital importance for the survival of these species. Interactions between agents in a population can be viewed as instances of evolutionary game theory. Each interaction places two agents together whereby one agent needs the other to contribute some resource to them. Here a resource is defined in an abstract sense as any kind of helpful act that contributes to the chance for survival of a peer. This is an altruistic act of the individual, but a requirement for the survival of the entire population.

1.1. The Evolution of Social Cooperation

Natural selection engenders competition among agents in a population such that selfish behaviour is rewarded. This can lead to a dilemma, commonly known as the "tragedy of the commons", in which the incentives of the individual are not aligned with those of the population as a collective. Such a dilemma is partially thwarted by the evolution of a number of mechanisms that induce cooperation in a population. Without any mechanism for the evolution of cooperation, natural selection favors defectors, which consequently outlive honest agents until there are only defectors left. Research has shown that there are 5 predominant mechanisms that biological communities adopt in order to maintain cooperation, namely *kin selection, direct reciprocity, indirect reciprocity, network reciprocity* and *group selection*. The idea behind these particular mechanisms is to reward behaviour of individuals that is beneficial to members of the population other than themselves and to some degree even punish "selfish" behaviour [15].

Agents in a population will incur some cost for performing an altruistic act, while the recipient will receive some benefit. Different mechanisms of social cooperation will yield different cost and benefit functions for altruisitc acts, based on which agents will decide whether or not it is sensible for them to cooperate. Nowak (2006) have determined under which restrictions of cost and benefit, cooperation will naturally evolve in biological communities.

- Kin Selection: Natural selection can favour cooperation if contributer and beneficiary are genetic relatives. Such an act is beneficial if the cost-to-benefit ratio exceeds the factor of relatedness, whereby the factor of relatedness is determined by the probability that both interaction partners share a gene.
- Direct Reciprocity: In the case of repeated encounters between the same individuals with consecutive rounds of interactions, an agent will decide whether to cooperate based on its contender's previous action. The most common form of direct reciprocity is known as the tit-for-tat strategy in game theory, which we will elaborate on later. Direct reciprocity leads to global cooperation if the cost-to-benefit ratio is exceeded by the probability of another interaction between the same agents.

- Indirect Reciprocity: In indirect reciprocity a node cannot rely on reencountering one of its previous interaction partners, but instead is likely to encounter strangers over and over again. This necessitates a mechanism that works on the basis of reputation. Agents may not have the chance to reciprocate directly. Instead, people contribute to the community on the assumption that it will increase their reputation, which in turn increases the probability of them receiving some work from a stranger. This mechanism only induces cooperation if the probability of knowing someone's reputation exceeds the cost-to-benefit ratio of the interaction.
- Network Reciprocity: We can picture this best in a graph-theoretical setting, in which every agent pays a cost for all of their neighbours in the social graph to receive a benefit. If they defect then their neighbours don't receive a benefit. Cooperators can prevail by forming network clusters among themselves, by only interacting with cooperators. This rule leads to cooperation if benefit-to-cost ratio exceeds the average number of neighbours each node has.
- Group Selection: The network is subdivided into groups and these groups grow as offspring is produced. As a group reaches a certain size it splits in two and another group is eliminated. We find that defectors in a mixed group proliferate faster than honest nodes. However, groups consisting of only honest nodes split much faster than mixed groups or groups with only defectors. Hence honest groups soon dominate the network. This only happens provided that the benefit-to-cost ratio is greater than the ratio of all nodes to number of groups plus 1.

Out of all biological species there are on this planet, the human race has developed the by far most sophisticated and effective mechanism to enforce indirect reciprocity across its entire population; Language. While many other life forms on earth have developed ways of communicating with one another, humans have developed the most intricate and complex method of communication. This is the reason that from an evolutionary perspective, we have arguably outdone all other biological species on this planet. Language enables humans to *gossip* about one another. While this might not initially seem like a significant contributor to reproductive success, it allows humans to share information about the reliability of their peers and the likelihood that an individual will act cooperatively in the future. Based on this shared information humans can cultivate a reputation which reflects their standing in a population.

This reputation mechanism rewards altruistic behaviour and punishes uncooperative acts. If, in an interaction with a peer an agent decides to defect then that peer will spread information about the agent's defection and if said agent has another interaction with a new peer that knows about its past defection then the agent is less likely to be collaborated with. Humans have developed an awareness of their own reputation over time which prompts them to behave cooperatively most of the time. Even with strangers whose reputation they might not know, humans often act politely and considerately, due to this awareness. While kin selection and direct reciprocity can ensure the cooperation of smaller tribes and families, reputation is a key element in the functioning of large-scale societies.

The upshot here is that bad behaviour is punished with bad reputation, while good behaviour is rewarded with good reputation. Good reputation leads to trust between two individuals, which results in more effective cooperation between individuals. Lastly, humans have incorporated "forgiveness" into this scheme. Individuals' reputations are malleable and dynamic. Agents that have misbehaved and developed a bad reputation can change their behaviour and redeem themselves, correcting their wrong-doings and fixing their reputation. This upgrades the reputation mechanism and ensures that defectors are incentivised to rectify their strategy and become cooperators.

1.2. Cooperation and Behaviour on the Internet

With the advent of one of the most disruptive technological revolutions in human history, namely the Internet, humans have been given an entirely new platform to interact on globally. There are a wide variety of different networks in which different types of resources are shared, from P2P file sharing networks, where agents up- and download data to one another to social networks where humans interact by sharing content with one another and rewarding or chastising it with "likes" or "retweets", etc. The social graph of human interaction has changed and especially grown significantly with the help of these tools. This changes the paradigm of human interaction and consequently their behaviour. It has been commonly observed that people are often much "nastier" to one another on the Internet than they are in real life. This nastiness can be considered as a form of defection against paradigms of social interaction, leading us to believe that the aforementioned rules for inducing cooperation no longer function effectively.

On the Internet humans no longer interact face-to-face and, more importantly, no longer need to disclose their identity to one another. Identity, however is an indispensable input of any reputation mechanism. For a reputation mechanism to be effective, identities need to be permanent and unique. When humans have the ability to hide their identity behind one or several pseudonyms, they can defect without having to face any long-term repercussions. Malicious peers may hide behind one or several pseudo-identities or even erase previous identities entirely in order to avoid bad reputation as a result of bad behaviour. The regular mechanisms for cooperation are no longer applicable and a new online analogue to reputation might have to be devised. This problem becomes particularly apparent in online social networks.

Social networks such as Facebook and Twitter, etc. struggle to clamp down on malicious behaviour such as cyberbullying and the proliferation of hateful content or "fake news". In the physical world this type of behaviour would be strongly disincentivised by the mechanisms of cooperation given in 1.1. A bully for instance, will be socially frowned upon and become an outcast from the community if their behaviour is not rectified. Of course, even in the real world these cooperative mechanisms are not implemented perfectly, however they do work. On the Internet, due to the reasons discussed above, we find that this is no longer the case. Reputation is no longer a reliable piece of information and trust is harder to achieve, making these networks increasingly uncooperative environments. While companies running these networks do their best in utilising technology to prevent and mitigate bad behaviour, they have so far not succeeded entirely.

1.3. Cooperation in Peer-to-peer Filesharing Networks

Another preeminent setting in which this problem of cooperation arises are online P2P filesharing networks. Peer-to-peer file sharing refers to the distribution of digital media over a P2P network without the need for any central authority or database. Files are located on individuals' computers and shared with other members of the network through up- and downloading data to one another. P2P software was the piracy method of choice in the early 2000s with software programmes such as LimeWire, Gnutella and the BitTorrent client being the most prominent applications [32]. A Supreme Court decision in 2005 led to the closure of many of these sites for illegally sharing copyrighted material. However, these applications are still very much in use today.

In P2P file sharing networks agents up- and download files amongst one another through acts called *seeding* and *leeching*. Agents holding a particular file will receive requests for the given file they are holding by so-called leechers. Nodes that require a particular file join a swarm of other nodes with the same needed file. Agents willing to seed now have to decide who to make a contribution to. The file they are willing to share is split into smaller pieces which are distributed among members of the swarm in a manner that ensures a fair distribution and prevents data from going extinct when agents go offline or leave the mesh.

P2P filesharing networks are an instance of computing distributed systems, which do not have any central authority governing the network. Instead of connecting to a central server for data, agents interact freely in a decentralised manner as visualised in figure 1.1.

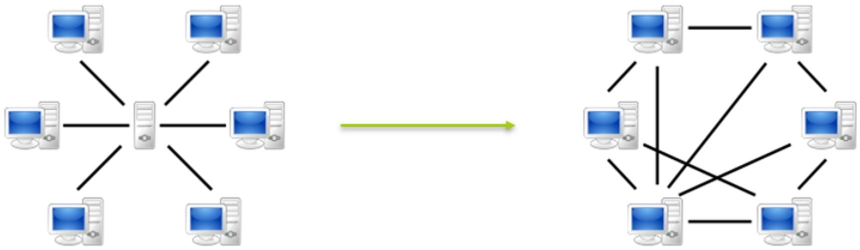


Figure 1.1: Client-Server vs P2P Model

There are some advantages and some disadvantages of the distributed nature of P2P networks over the traditional client-server model and their applicability depends on the context. The most notable are listed in the table given in figure 1.2 below.

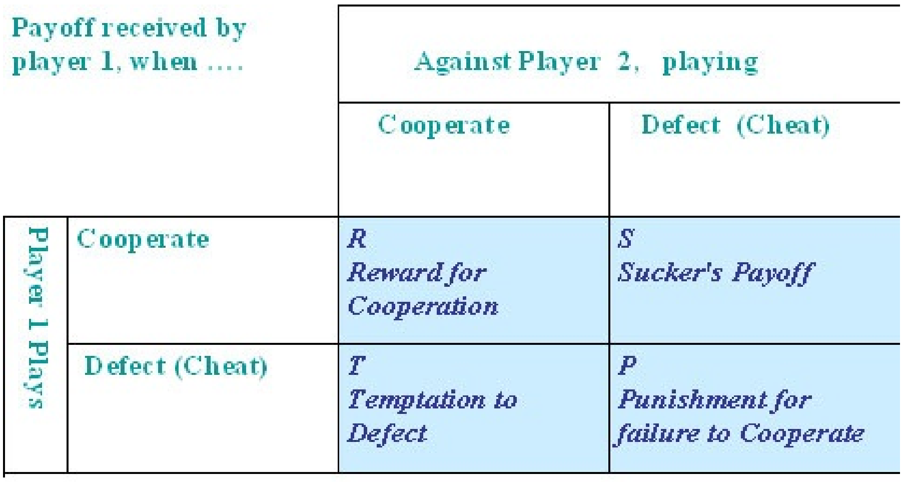
Advantages:	**Disadvantages:**
No single point of failure No network congestion No expensive server architecture needed	No Accountability Possible malware on the network No backup of data Updates are difficult to implement

Figure 1.2: Advantages and disadvantages of P2P over Client-Server

In most online networks with some kind of central authority, such as Ebay, Airbnb, etc. cooperativeness is achieved through review mechanisms, which are maintained and secured by the central authority. Agents can evaluate the trustworthiness of their potential interaction partners, by assessing their previous transactions and other agents' opinions of them. These reviews are stored on a central database which the central authority maintains. Seeing as the point behind P2P networks was to eliminate this central authority the problem of cooperation arises again. Users have an obvious incentive to download, but no inherent incentive to share data. This is what we referred to earlier as *the tragedy of the commons*, which results in behaviour we call *lazy freeriding*, where agents leech excessively, but do not seed. Different file sharing platforms have different mechanisms to enforce the necessary altruistic sharing of files.

1.3.1. BitTorrent & Direct Reciprocity

To facilitate cooperation the most prominent P2P network, BitTorrent, employs a mechanism called *tit-for-tat*, which is an instance of direct reciprocity. Tit-for-tat is a highly effective strategy in game theory for the iterated prisoner's dilemma, in which an agent cooperates first and then replicates its contender's previous actions as seen in figure 1.3. In practice, this works as follows. Peers in the BitTorrent network have a limited number of upload slots to allocate. An agent will begin by exchanging upload bandwidth for download bandwidth with a number of its peers. If one of these peers turns out to be a leecher, i.e. does not reciprocate, it will be choked out. This means the agent will discontinue it's cooperation and assign the corresponding upload slot to another randomly chosen peer in a procedure known as *optimistic unchoking*.



Payoff received by player 1, when		Against Player 2, playing	
		Cooperate	Defect (Cheat)
Player 1 Plays	Cooperate	*R* *Reward for Cooperation*	*S* *Sucker's Payoff*
	Defect (Cheat)	*T* *Temptation to Defect*	*P* *Punishment for failure to Cooperate*

Figure 1.3: Instance of the Prisoner's Dilemma, in which Tit-For-Tat is the Dominant Strategy. Image taken from [33].

However, we find that in the case of fleeting and asymmetric interactions, tit-for-tat is no longer very effective [16]. Fleeting and asymmetric means that agents have many unrepeated and unreciprocable interactions with their peers. When there is a high probability of two agents not seeing each other again, peers cannot be evaluated based on their previous reliability and hence every new transaction entails the risk of the contender defecting. In tit-for-tat agents do not keep a memory about their peers' reliability and do not share information about this behaviour with the network. In such a setting defecting becomes the dominant strategy of the Prisoner's dilemma [3]. The agents' inability to coordinate and build expectations of their counterparts ensures that defection will rarely be punished. Everyone is worse off than if they had collaborated, but no individual can gain anything by changing to a collaborative strategy, since there's almost never a reward. This is what we referred to as the tragedy of the commons earlier in section 1.1. We find that a mechanism of indirect reciprocity may be more successful at inducing cooperation in these types of networks.

1.3.2. Tribler & Indirect Reciprocity

The *Distributed Systems Group* at Delft University is running and developing an open-source P2P file sharing network, called *Tribler*, which aims to leverage the power of mechanisms for social cooperation in an attempt to create a more reliable file sharing platform. It is designed with a custom built-in onion routing network whereby the transference of data is routed through several relay nodes before reaching the leeching node, ensuring anonymity of participants and clients can participate in any BitTorrent network. Tribler is trackerless and built on an overlay network for content searching, rendering it truly decentralised and immune to limiting external action such as government restraint.

> Johan Pouwelse. „The only way to take down Tribler is to take down the Internet." (Dailymail 2009)

In an attempt to alleviate the problem of freeriding, Tribler aims to incorporate mechanisms of *indirect reciprocity* to enforce cooperation. Agents gossip about their transaction partners and inform others about their trustworthiness. Agents' respective transaction histories are disseminated along the network. From this information agents can aggregate an approximation of their peers' reputations such that freeriders and otherwise uncooperative agents can be identified. An agent that holds parts of a particular file will receive queries from peers that require that particular file just like in the BitTorrent protocol. The agent holding the file will then decide whom to upload to, based on the reputation of the nodes in the swarm. After having some work performed the reputation of the performer should increase while that of the recipient should decrease, such that in the next interaction the peer that has performed the work will have a higher probability of receiving work and the recipient will have a lower one. Uncooperative nodes are therefore not completely shunned, but are restrained in their ability to consume data.

1.4. Blockchains and TrustChain to Enhance Online Cooperation

In order for agents to be able to evaluate their peers' reputation based on their respective transaction histories there needs to be a database logging all agents' interaction histories. However, seeing as it's Tribler's goal to avoid any kind of centralisation, a distributed storage, or ledger, is required. The most commonly used tool for this purpose are Blockchains. Blockchains are append-only data structures that utilise cryptographic primitives such as public-key cryptography and digital signatures to maintain a consensus on data, stored on many different processors in a distributed system. Transactions between agents in the network are grouped in blocks which, in turn, are interlinked by a hash chain.

The most popular type of blockchain is given by the Bitcoin proof-of-work blockchain, in which blocks are created by "miners"; nodes in the network that collect and group transactions. In order to obtain a block, the miner needs to solve a crpytographic hash puzzle through a protocol known as proof-of-work (PoW). If conflicting states occur, the chain forks, and miners contribute to the chain they believe is the valid one. At some point, one chain will overtake the other and all miners transition to *that* chain. This point is determined by a certain number of blocks by which one chain surpasses the other, which is based on a predetermined lower bound for the probability of a dishonest miner single-handedly overruling the current chain. The resulting chain of blocks is therefore immutable as well as fraud-proof. The idea behind behind PoW and miners is that authority to make changes to the log is randomised, making it impossible for any single agent to obtain any significant authority over what is stored on the Blockchain [13].

Blockchains however have a major drawback that the classical client-server model does not have. In order to ensure randomisation of append-authority and transaction validity agents are required to wait for a certain number of blocks to exceed a transaction's block before this transaction is deemed valid. This fundamentally limits their scalability in terms of transaction throughput. In pursuing a more scalable alternative, the distributed systems group of the TU Delft has developed their own type of distributed ledger, called TrustChain [20]. TrustChain is what is known as a fourth-generation blockchain.

Unlike most traditional blockchains, all network participants maintain their own chain of transactions in the TrustChain protocol. There is no mining and no global consensus. The TrustChain maintains records of all interactions between peers in the network, in respective blocks. Blocks are linked to one another through hash pointers, whereby a block contains the hash value of its preceding block. Each block is thereby connected to two preceding and two succeeding blocks, i.e. each block is contained in the chains of both transaction partners. This results in many interlinked chains, each corresponding to a single agent's transaction history. For a visualisation of the TrustChain datastructure see Figure 1.4.

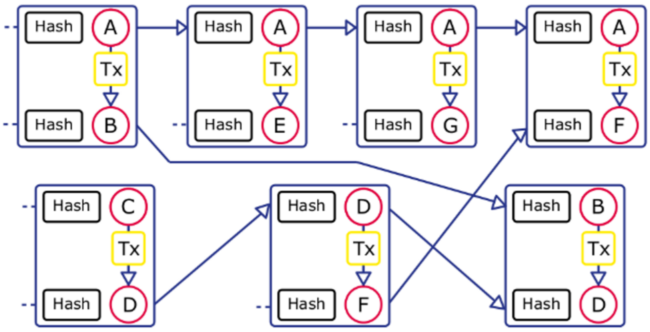


Figure 1.4: TrustChains of different network participants (taken from [20]).

This structure is strongly scalable, both in the number of agents in the network as well as in the number of transactions per agent as TrustChain does not maintain a global consensus. This means that double-spend attacks are not actually prevented, as they are in traditional blockchains. However, they are made detectable through a gossip-protocol, as peers share information about other nodes' transaction histories and can subsequently be penalised. Thereby fraudulent activity is not actually prevented, but strongly disincentivised.

2 Research Question

The Tribler P2P network aims to incorporate a reputation mechanism into their application to enforce indirect reciprocity and thereby achieve cooperation. Ultimately, the goal is to determine an algorithm that takes the TrustChains of agents participating in the network and returns some reputation scores for these nodes. Reputation is subjective and therefore reputations should be determined by all nodes independently based on the data they have gathered through the gossip protocol. There are many algorithms that spring to mind that may achieve a desirable outcome in this setting. However, designing such an algorithm comes with a particular set of challenges that must be overcome for it to be effective. This leads us to our research question:

What requirements does a reputation mechanism need to satisfy to induce cooperative behaviour in a P2P file sharing network?

In order to answer this question we begin by refining our understanding of a reputation mechanism. In [23] Seuken & Parkes (2011) introduce the concept of *accounting mechanisms* which are mappings on a social interaction graph representing the reputability that agents in the P2P network have based on their interaction histories. If one agent is assigned a higher score than another agent by this mapping then that agent is considered more reputable. The main idea is that a sensible accounting mechanism should assign higher scores to nodes who make overall larger contributions to the network and consume less than other nodes. A node with a higher reputation score should then find itself more likely to be served data, therefore stimulating cooperative behaviour. Conversely, agents that behave selfishly should be assigned lower reputation scores and should therefore be less likely to receive data, disincentivising selfish behaviour.

Ideally, accounting mechanisms should entail some transitivity, by which we mean nodes assign agents that they have had direct interactions with higher reputation scores than nodes they have not had direct interactions with. The larger the distance between two nodes in the social interaction graph the smaller the reputational reward for a contribution. Additionally, contributions that are indirect contributions to a node should lead to higher reputation scores than contributions that are not indirectly benefitting this node. By indirect contributions we mean that if a node contributes some resources to another node which in turn serves a third node, then the third node will consider the contributions made by the first node indirect contributions to itself. This would accurately capture the concept of reputation as encountered in the real world.

Lastly, an accounting mechanism should successfully prevent **lazy freeriding**. We consider an agent that consumes far more resources than they contribute a lazy freerider. More rigorously, we say that if the net contributions a node has made to the rest of the network, i.e. the amount they have consumed subtracted from the amount they have contributed, exceed a certain lower bound then that node is a lazy freerider. Alternatively, we might say that if the ratio of these values exceeds a given lower bound then that node is a lazy freerider. An accounting mechanism should penalise excessive leeching in a manner that makes it impossible for a node to go below such a threshold.

So far, this question seems like a rather easy one to solve. There are plenty of algorithms that will capture these requirements and prevent lazy freeriding. However, the question is complicated by the possibility of attacks on the file sharing network. In this work we will focus on two types of attacks in particular, namely misreport attacks and sybil attacks.

A **misreport attack** is performed by one or more malicious agents who do not report honestly on their own past interactions. Malicious agents may try to deceive honest agents by reporting on transactions that have not actually occurred or by concealing transactions that may reduce their standing in the network. By this method agents can increase their reputation or reduce the standing of other nodes in the network. In [24] Seuken & Parkes (2011) have introduced a mechanism which solves this problem to some degree. In this work we examine the ability of the TrustChain architecture to prevent this type of attack.

A **sybil attack** occurs when a single malicious agent creates multiple, often times large amounts of, fake identities. This agent will then attempt to exploit the control they have over the accounts in order to artificially increase the reputation score of one or more of their identities by reporting high levels of reputability through fake transactions without actually performing any work. Another approach may be to simply reduce the reputation of other honest nodes in the network to improve their own relative standing(s). This can be done because Sybil identities can create forged reports about one another. Such attacks can have strongly detrimental effects on the functioning of P2P networks, especially if carried out on a large scale. If the creation of identities and forging of transactions are cheap compared to their gain, then such attacks have the potential to disrupt entire file-sharing networks.

Given these types of attacks we can narrow down our research question to the following

What requirements does an accounting mechanism need to satisfy in order to effectively incentivise cooperation and prevent lazy freeriding, while being resistant to misreport attacks and mitigating the effects of sybil attacks?

2.1. Thesis Summary and Contributions

In chapter 3 we begin by mathematising the relevant concepts such as transactions, work graphs, accounting mechanisms, allocation policies, lazy freeriding, misreports, sybil attacks and the TrustChain architecture. We prove the resistance of accounting mechanisms that are based on the TrustChain architecture to misreports under some mild restrictions and we prove the resistance of certain types of accounting mechanisms to lazy freeriding. This chapter is meant to simply provide a framework in which we can conduct our research.

In chapter 4 we elaborate on the effects of sybil attacks. We introduce the concept of its cost and profit for the attacker. The cost of a sybil attack turns out to be easily explained, while defining the profit turns out to be more involved. We solve this problem by postulating an interaction model in which we also evaluate which allocation policies are most resistant to sybil attacks. Given this model we can determine a formula for the profit of a sybil attack. This is the amount of additional work that can be consumed by the attacker after the attack has been carried out. Seeing as this formula is based on a discrete stochastic process we realise that it is impossible to compute in a generic setting. In order to obtain values for the cost and profit of a sybil attack that are practically computable we redefine these values for accounting values, i.e. the aggregate of additional accounting values obtained by the attacker and its sybils. We believe that rigorous definitions of these terms are much needed and have been neglected in the existing literature such as in [23].

The values of sybil attack cost and profit in terms of accounting values are now much easier to compute. However, they are not actually the relevant metric, but just a proxy for the earlier defined cost and profit of sybil attacks in terms of work. Given these two different definitions we investigate the relationship between them. We introduce examples where the two above are not equivalent in chapter 5. This turns out to be very problematic indeed as accounting mechanisms only serve as a representation of a node's cooperativeness and a sybil attacker aims to increase these in an attempt to obtain more work from the network. Therefore we find that some accounting mechanisms do not allow for accurate assessment of sybil attack profit. In order to circumvent this dilemma we come up with the definition of representativeness to ensure consistency between these two concepts of sybil attack profit.

In chapter 6 we analyse existing impossibility results from the literature which state under which conditions accounting mechanisms are susceptible to sybil attacks that enable the attackers to consume large amounts of data [23]. We detect an error in an important theorem and extend the existing model to circumvent this error. We introduce two additional properties of accounting mechanisms which ensure the existence of impactful sybil attacks and produce two further impossibility results as well as consequent corollaries for slightly relaxed versions of the two properties.

In our last chapter, chapter 7 we aim to do the inverse of what we did in chapter 6, i.e. introduce properties for accounting mechanisms to be resistant to strongly beneficial sybil attacks. We begin by characterising certain types of passive sybil attacks, namely parallel and serial attacks. Next, we introduce requirements for accounting mechanisms to be resistant to these types of attacks. We extend the model to a particular type of sybil attack to which accounting mechanisms that are resistant to the upper types of attacks, are also resistant. Lastly, we extend our requirements for accounting mechanisms by an aditional property to be obtain resistance to arbitrary types of sybil attacks as well.

In the appendix in chapter A we address research we conducted that did not turn out to be fruitful. The first approach to solving this problem was through a model based on geographic proximity of participants in the network. The second topic we analysed was the topic of the evolution of cooperation among biological organisms. For this we made a month long research visit to Japan to analyse properties reputation mechanisms should satisfy to be able to facilitate cooperative behaviour.

3 Mathematical Framework for Accounting Mechanisms

We begin by introducing a mathematical framework for the setting in which we conduct our research, namely by rigorously formalising interactions (transactions) between nodes in the network. In [19] Otte et al. (2016) introduced the concept of an *ordered interaction model* from which an *ordered interaction graph* and a *block graph* are derived. While this is a very elegant definition for a set of transactions and the derivation of a work graph from it, it is directly tailored to the TrustChain architecture and lacks the possibility of misreports and counterfeit interactions. Therefore we will not adopt it here, but instead derive a slightly different and more generic definition of a transaction set, which will be our equivalent to their ordered interaction model.

3.1. Network Transactions

We start off with the definition of a simple network transaction, or interaction, which simply denotes the transference of data in between two nodes.

Definition 3.1.1 (Agent Transaction)**.**
Let V be the set of all agents in the network and let pr_1, pr_2, pr_3 denote the canonical projections on the cartesian product of 3 sets. A transaction $t \in V^2 \times \mathbb{R}_{>0}$ between two nodes $i, j \in V$ is given by a tuple (i, j, w), whereby $pr_1(t)$ is the contributer and $pr_2(t)$ is the recipient of the work performed. w or $pr_3(t)$ corresponds to the size of the transaction, i.e. the amount of data transferred from $pr_1(t)$ to $pr_2(t)$.

Note that for any transaction t it must always hold $pr_1(t) \neq pr_2(t)$, i.e. nodes cannot transact with themselves. Secondly, transactions are unidirectional. This means that a single transaction cannot contain the transference of data from node i to node j **and** vice versa. Hence, the ordering of the two nodes in the transaction tuple is not arbitrary, but determined by which of the two is making the contribution and who is receiving it. Lastly, it naturally always holds that $w \geq 0$.

As every node participates in a string of transactions in a given chronological order, we obtain a series of transactions for every node i, which we will refer to as a transaction sequence.

Definition 3.1.2 (Transaction Sequence)**.**
The transaction sequence of a node $i \in V$ is expressed as $TS_i := (t_{i,n})_{n \in \mathbb{N}_{\leq T_i}}$ where $t_{i,n}$ is the n-th transaction node i participated in, either as a contributor or as a consumer. As above $t_{i,n}$ is given by a tuple (j, k, w) where either $j = i$ or $k = i$. T_i denotes the length of i's transaction sequence, i.e. the number of transactions i has

participated in thus far. It grows as time progresses.

Note that in this definition we implicitly assume that concurrent transactions can be deterministically serialised. Else, the ordering of transactions would become nonsensical. As time goes on, transaction sequences obtain new entries and continue to grow, which implies that T_i is not a static value, but changes over time. We choose not to incorporate a temporal variable in this model and instead assume that a transaction sequence represents a "snapshot in time" as opposed to a dynamic variable. Next, we define a transaction function, which will denote the size of a transaction.

Definition 3.1.3 (Transaction Function)**.**
For every node $i \in V$ we define a transaction function t_i, given by

$$t_i : \mathbb{N}_{\leq T_i} \times V \to \mathbb{R},$$

where $t_i(m, j)$ corresponds to the amount of work node i has leeched from or contributed to node j in its m-th transaction, i.e.

$$t_i(m, j) = \begin{cases} pr_3(t_{i,m}), & \text{if} \quad pr_2(t_{i,m}) = j \\ -pr_3(t_{i,m}), & \text{if} \quad pr_1(t_{i,m}) = j \\ 0, & \text{otherwise} \end{cases}.$$

Note that it holds

$$t_i(m, j) > 0 \ \text{ if } pr_1(t_{i,m}) = i$$

and

$$t_i(m, j) < 0 \ \text{ if } pr_2(t_{i,m}) = i.$$

It is obvious that the two conditions above can never both be satisfied simultaneously. This is due to our restriction made in definition 3.1.1, where we stated that for any transaction t it must always hold $pr_1(t) \neq pr_2(t)$.

Remark 3.1.1 (Symmetry of Transaction Functions).
In theory it should always hold for any pair of nodes $i, j \in V$ and any value $w \neq 0$

$$\left|\{n \in \mathbb{N}_{\leq T_i} \mid t_i(n, j) = w\}\right| = \left|\left\{m \in \mathbb{N}_{\leq T_j} \mid t_j(m, i) = -w\right\}\right|.$$

What this means is that any transaction between nodes i and j that is contained in the transaction sequence of i must also be contained in the transaction sequence of node j. This is quite trivially true if the transaction sequences of both parties contain all transactions that they have participated in. We call this property *symmetry of transaction functions.*

Finally, we introduce the set containing all transactions that have transpired in the network, denoted by

$$TS := \{TS_i \mid i \in V\}.$$

This set contains all transaction sequences of all nodes in the network. Based on our remark 3.1.1 we see that TS must contain every transaction exactly twice.

Recall that in a distributed system there is no central authority and therefore no central database keeping record of all transaction sequences. Hence, an agent can only know their own transaction sequence and those of agents who've shared their transaction sequences with them. Agents are unlikely to be aware of the transaction sequences of all agents in the network, or of who is in the in the network in the first place. Hence no agent can know the transaction set TS.

Agents query one another's transaction sequences which are then shared and disseminated along the network. In [8] Harms et al. (2018) propose a record dissemination protocol, which is based on the TrustChain architecture, discussed in section 1.4. We will not delve into the details of mechanisms facilitating this distribution of transaction sequences, but will simply assume that there is one in place and continue.

Definition 3.1.4 (Agent Information)**.**
Let $i \in V$ be an arbitrary, but fixed agent in the network. An interaction (j, k, w) between two agents $j, k \in V$, that i receives a report about from j is written as $t^i_{j,m}$, where the m means that it's the m-th transaction in the transaction sequence j has shared with i. The transaction sequence j reports to i is then denoted by $TS^i_j := (t^i_{j,m})_{m \leq T^i_j}$. Here T^i_j is the length of TS^i_j. We derive the transaction function of j that i has information on as

$$t^i_j(m,k) = \begin{cases} pr_3(t^i_{j,m}), & \text{if} \quad pr_2(t^i_{j,m}) = k \\ -pr_3(t^i_{j,m}), & \text{if} \quad pr_1(t^i_{j,m}) = k \\ 0, & \text{otherwise} \end{cases} .$$

When aggregated into a set of all transaction sequences, i obtains the *subjective* transaction set

$$TS^i := \left\{ TS^i_j \mid j \in V \right\}.$$

Recall that TS contained every transaction exactly twice. This is no longer true for TS^i as different agents may report transaction sequences inconsistently. This means agent information may be contradictory or flawed.

So far, we have not ensured that agents sharing their transaction sequences will do so honestly and consistently. Agents may choose to add transactions that haven't occurred to their transaction sequence or drop transactions from their sequence. Agents may even refuse to share their transaction history entirely. This type of behaviour is what we defined earlier as *misreports*, which we will define more rigorously now.

Definition 3.1.5 (Misreport Attack)**.**
Let $i \in V$ be an arbitrary but fixed agent with subjective transaction set TS^i. We say that a misreport between two agents $j, k \in V$ has occurred if there exists a $w \neq 0$ such that

$$\left| \left\{ n \in \mathbb{N}_{\leq T^i_j} \mid t^i_j(n,k) = w \right\} \right| \neq \left| \left\{ m \in \mathbb{N}_{\leq T^i_k} \mid t^i_k(m,j) = -w \right\} \right|.$$

Put in words this simply means that a misreport between two agents j and k has occurred if there exists an agent i who receives reported transaction sequences TS^i_j and TS^i_k such that the there exists a transaction between j and k, which is contained in the reported transaction sequence of one of the two, but not in both.

Note that we say a misreport *has occurred* instead of a misreport was *committed by* as it is not clear to agent i which of the two agents j and k is responsible for the misreport. If a transaction is contained in the reported transaction sequence of agent k and not in that of j, then either k may have fabricated a transaction or j may have dropped a transaction from their sequence. From the perspective of i these two cases are indistinguishable.

3.2. Work Graphs

Given the set of all transactions TS, one can transform the transaction sequences into a *work graph* with the help of a mapping function. A work graph is a directed network graph visualising the interactions between nodes. It may be unidirectional or even a double-edged graph. The idea is that edges between vertices correspond to overall seed-leech relationships of nodes in the network.

Definition 3.2.1 (Work Graph)**.**
A work graph is given by the tuple $G = (V, E, w)$, whereby V is the set of vertices, i.e. agents in the network and E is a set of directed edges between the agents. An edge $(i, j) \in E$ pointing from node i to node j represents node j performing work for node i.

The function $w : V \times V \to \mathbb{R}_{\geq 0}$ denotes the weight of the edges, i.e. $w(i, j)$ represents the total amount of work performed by node j for node i. If two nodes i and j are not connected then we set the edge weights $w(i, j) = w(j, i) = 0$. We choose the set of edges $E \subset V \times V$ such that $(i, j) \in E$ if and only if $w(i, j) > 0$. Note that it must always hold $w(i, i) = 0$ f.a. $i \in V$ as we do not allow for agents to transact with themselves.

There are a number of different ways transactions in TS can be aggregated to form edges in the work graph. In the *unidirectional, single-edge* case of the work graph the edges of the graph can be derived from the set of transaction functions by

$$w(i, j) = \max\left\{\sum_{n \in \mathbb{N}_{\leq T_j}} t_j(n, i), 0\right\} = \max\left\{-\sum_{n \in \mathbb{N}_{\leq T_i}} t_i(n, j), 0\right\}$$

and conversely,

$$w(j, i) = \max\left\{\sum_{n \in \mathbb{N}_{\leq T_i}} t_i(n, j), 0\right\} = \max\left\{-\sum_{n \in \mathbb{N}_{\leq T_j}} t_j(n, i), 0\right\}.$$

Here the weight of the edges corresponds to the net data flow in between two nodes. The edge is directed toward the node that has a positive deficit in the bilateral relationship. Note that there can only be a single edge connecting two nodes, which points from one to the other, i.e. for any pair of nodes $i, j \in V$ it holds $w(i, j) > 0 \Rightarrow w(j, i) = 0$. This type of work graph is quite useful as it nicely captures the overall net contributions nodes have made to the network. Another advantage this type of work graphs has is its simplicity as there is never more than one edge connecting two nodes.

Note that there is one drawback to this approach, which lies in the fact that the single-edge graph neglects certain contributions made to the network. For instance, if two agents have donated the same amount of resources to one another then the weight of the edge connecting them is zero. Hence, this type of graph lacks informativeness as it only captures net contributions.

An alternative to this are double-edged graphs in which case, we can derive the edge weights as follows

$$w(i, j) = \sum_{n \in \mathbb{N}_{\leq T_j}} \max\{t_j(n, i), 0\} = \sum_{n \in \mathbb{N}_{\leq T_i}} \max\{-t_i(n, j), 0\}$$

and

$$w(j, i) = \sum_{n \in \mathbb{N}_{\leq T_i}} \max\{t_i(n, j), 0\} = \sum_{n \in \mathbb{N}_{\leq T_j}} \max\{-t_j(n, i), 0\}.$$

In this particular type of graph an edge (i, j) corresponds to the gross data flow from j to i without subtracting the work i has done for j. A positive attribute of this this type of graph is that it's generally more informative as it doesn't reduce edge weights to net data flow. Throughout this thesis we will always assume a double-edged work graph.

Given a transaction set TS, we can derive this type of work graph using the mapping function mentioned above. We write $G = g(TS)$ where g maps the transaction set TS to the work graph G, according to the classifications above.

It may be somewhat counterintuitive for edges to be pointing from the recipient to the contributor. Note that we can invert the edges as well, such that an edge (i, j) pointing from i to j corresponds to work performed by i for j, in that case we obtain.

$$w(i,j) = \sum_{n \in \mathbb{N}_{\le T_i}} \max\{t_i(n,j), 0\} = \sum_{n \in \mathbb{N}_{\le T_j}} \max\{-t_j(n,i), 0\}.$$

However, we choose to stick with the former direction with a particular set of accounting mechanisms in mind. Although, this is quite irrelevant. For an example of how to derive a work graph from a transaction set see the example below.

Example 3.2.1.
Take the tabular below as the transaction sequences of 4 agents $i, j, k, h \in V$. Then the mapping function g returns the corresponding work graph G with directed double-edges derived from the transaction set TS as seen in figure 3.1.

i	k	j	h
$(i,j,3)$	$(k,j,2)$	$(j,h,4)$	$(h,i,9)$
$(h,i,9)$	$(i,k,2)$	$(j,i,1)$	$(j,h,4)$
$(j,i,1)$		$(i,j,3)$	
$(i,k,2)$		$(k,j,2)$	

$$\Longrightarrow$$

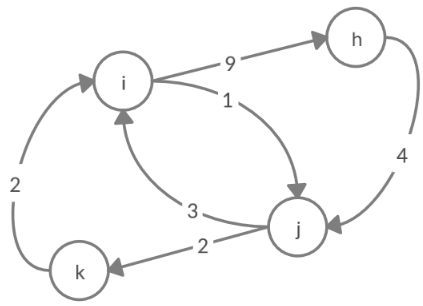


Figure 3.1: Example Work Graph

Remark 3.2.1.
Note that in our case we introduce the work graph with regard to peer-to-peer filesharing, keeping the application of the *Tribler* network in mind [21]. This means that the work performed, i.e. the weight of the edges, corresponds to the amount of data transferred from one node to another, by seeding and leeching respectively.

However, with our examples of social networks, such as Facebook and Twitter from chapter 1 in mind, we would like to extend our model to entail these networks and any kind of P2P network in general. In the case of these social networks we can apply the exact same concepts, but reinterpret the transactions between agents as "follow" or "friendship" relations, etc. The weights of these edges could then, for instance, be determined by the number of likes and/or retweets a user receives from a follower/friend.

Example 3.2.2.
In the case of Facebook a friendship may correspond to an undirected edge connecting two vertices, while the amount of likes and/or mentions these people receive from one another, could be represented by another pair of edges connecting the two. An example of such a work graph is given in figure 3.2 below.

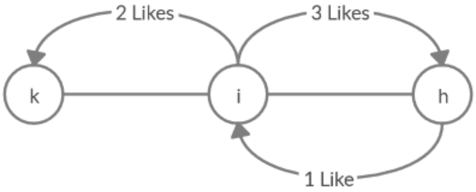


Figure 3.2: Facebook Example Graph

A follower relationship on Twitter could be represented by an edge pointing from the follower to the followed, while the edge weight may correspond to the number of tweets that have been liked or retweeted, etc. In this particular application a bidirectional graph will be more reasonable than a unidrectional one, as can be seen in figure 3.3 below.

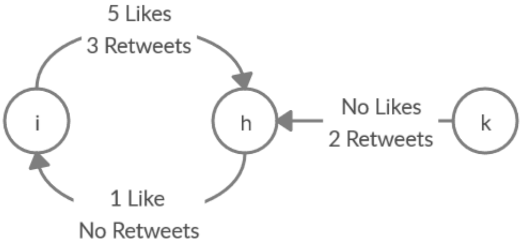


Figure 3.3: Twitter Example Graph

Recall the fact that agents in the network were not aware of all transactions that have occurred and definition 3.1.4, in which we stated that agents build a subjective transaction set based on agent reports. It follows from this that the work graph defined above is unlikely to be known by any node in the network. Instead, agents build, what is referred to, as a subjective work graph from their subjective transaction sets. This follows the same paradigm as above with one difference, which arises due to the possibility of misreporting.

As mentioned in definition 3.1.5 agents may report contradictory transaction sequences. This results in a work graph with edge weights given by tuples in $\mathbb{R}_{\geq 0} \times \mathbb{R}_{\geq 0}$, whereby one entry of the tuple corresponds to the aggregated data flow between the two nodes, as reported by one of them, while the other corresponds to the same value, but reported by the other.

Definition 3.2.2 (Subjective Work Graph)**.**
A subjective work graph from the perspective of node i is given by a tuple $G_i = (V_i, E_i, w_i)$ where $V_i \subset V$ and $E_i \subset V_i \times V_i$. As in the definition of the work graph an edge $(j,k) \in E_i$ pointing from j to k represents work performed for j by k.

For two nodes $j,k \in V_i$ the value $w_i(j,k)$ denotes the weight of the edge connecting j and k, as reported by both nodes in question to node i. Seeing as two nodes may report different transaction sequences, $w_i(j,k)$ is determined by a tuple $w_i(j,k) = (w^i_j(j,k), w^i_k(j,k))$. As before, if two nodes are not connected (from the perspective of i), we set $w^i(j,k) = 0 = w^i(k,j)$ and we choose the set of edges E_i such that $(j,k) \in E_i$ if and only if either $w^i_j(j,k) > 0$ or $w^i_k(j,k) > 0$. As in definition 3.2.1, we do not allow edges $w_i(j,j)$ for any $j \in V_i$.

The transaction sequences in TS_i can be aggregated into edge weights analogously to our earlier definition of the work graph (3.2.1). In the *unidrectional, single-edge* case the edge weights of the subjective work graph G_i are determined by

$$w^i(j,k) = \left(\max\left\{ \sum_{n\in\mathbb{N}_{\leq T^i_k}} t^i_k(n,j), 0 \right\}, \max\left\{ -\sum_{n\in\mathbb{N}_{\leq T^i_j}} t^i_j(n,k), 0 \right\} \right)$$

and consequently

$$w^i(k,j) = \left(\max\left\{ \sum_{n\in\mathbb{N}_{\leq T^i_j}} t^i_j(n,k), 0 \right\}, \max\left\{ -\sum_{n\in\mathbb{N}_{\leq T^i_k}} t^i_k(n,j), 0 \right\} \right).$$

Alternatively, we can aggregate the transaction sets TS^i into a *unidrectional double-edge graph* just as in the case of the work graph by setting

$$w^i(j,k) = \left(\sum_{n\in\mathbb{N}_{\leq T^i_k}} \max\left\{ t^i_k(n,j), 0 \right\}, \sum_{n\in\mathbb{N}_{\leq T^i_j}} \max\left\{ -t^i_j(n,k), 0 \right\} \right)$$

and

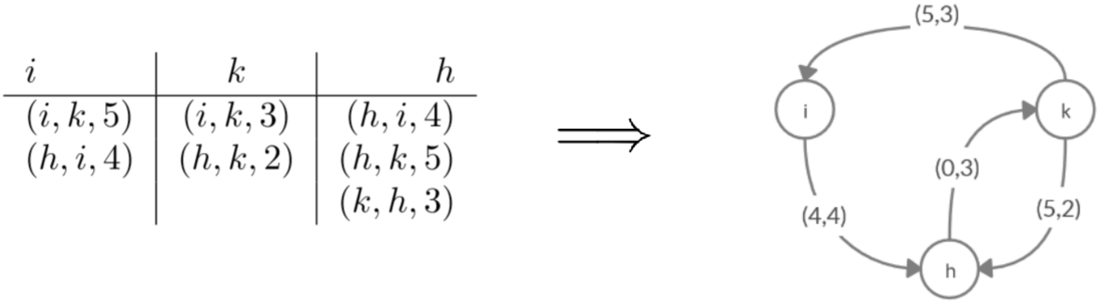
$$w^i(k,j) = \left(\sum_{n\in\mathbb{N}_{\leq T^i_j}} \max\left\{ t^i_j(n,k), 0 \right\}, \sum_{n\in\mathbb{N}_{\leq T^i_k}} \max\left\{ -t^i_k(n,j), 0 \right\} \right).$$

This results in every two nodes being assigned 4 values. If no misreport has occured in between two nodes j and k we replace the tuple of edge weights with a single value, as the tuple contains the same value twice, in which case we have $w^i(j,k) = w^i_j(j,k) = w^i_k(j,k)$.

Remark 3.2.2.
For the edges that are directly connected to agent i itself, i need not rely on the reports from the nodes it is connected to. It always knows with certainty the correct weight of these edges. Hence we find that $w_i(i,j)$ and $w_i(j,i)$ will be given by a single value as opposed to a tuple. We set $w^i(i,j) = w^i_i(i,j)$ and $w^i(j,i) = w^i_i(j,i)$.

When aggregating subjective transaction sequences into a subjective work graph, we apply a mapping function g_i and we write $G_i = g_i(TS^i)$. As mentioned above in definition 3.2.1, we opt for the multi-edge case for the same reasons as discussed above. Given below is an example of how a subjective transaction set TS^i can be aggregated into a subjective work graph.

Example 3.2.3.
Take the tabular below as the subjective transaction sequences of 3 agents $i,k,h \in V_i$ from the perspective of honest agent i. Then the map function g_i returns the corresponding subjective work graph G_i with directed double-edges derived from the transaction set TS^i as seen in figure 3.4 below.



i	k	h
$(i,k,5)$	$(i,k,3)$	$(h,i,4)$
$(h,i,4)$	$(h,k,2)$	$(h,k,5)$
		$(k,h,3)$

Figure 3.4: Example Subjective Work Graph

Remark 3.2.3.
If from the perspective of i no misreport has occurred between two agents j and k, the reported edge weights will satisfy $w^i_j(j,k) = w^i_k(j,k)$, which means $w^i(j,k)$ will be given by a single value. The question arises whether the occurrence of a misreport directly implies $w^i_j(j,k) \neq w^i_k(j,k)$.

We find that this is not, in fact true. Neither in the single-edge case, nor in the double-edge case. As a proof look at the following examples. In a single-edge graph assume j and k report the following transaction sequences to i

$$TS^i_j = ((j,k,5),(j,k,3),(k,j,1)) \quad and \quad TS^i_k = ((j,k,4),(j,k,3)).$$

Then the edge weights between nodes j and k in i's subjective work graph will be given by $w^i(k,j) = (7,7)$ and $w^i(j,k) = (0,0)$. Hence we have a misreport, but still it holds $w^i_j(j,k) = w^i_k(j,k)$ and $w^i_j(k,j) = w^i_k(k,j)$.

For the case of a double-edge graph we can think of a similar example with the same result. Let j and k report the transactions

$$TS^i_j = ((j,k,5),(j,k,3),(k,j,1)) \quad and \quad TS^i_k = ((j,k,6),(j,k,2),(k,j,1)).$$

In this case i will aggregate the transactions and obtain the edge weights $w^i(k,j) = (8,8)$ and $w^i(j,k) = (1,1)$.

Hence, we find that the occurrence of a misreport does not directly imply an unequal pair of reported edge weights. When one keeps in mind the fact that agents misreport with the intention to make themselves appear more cooperative it may seem somewhat counterintuitive for two agents to perform misreports which will yield the same edge weights, as lying about the value of an edge weight always makes one of the two nodes appear more and the other less altruistic. However, there are cases in which there exists an incentive for such misreports to occur.

These types of misreports are however invisible in the subjective work graph as the edge weights are the same. They are therefore impossible to detect from only looking at the subjective work graph. Later on we will introduce two mechanisms of misreport-prevention, one of which can prevent this type of misreport. If we limit our scope to misreports that are detectable and visible in the subjective work graph we can introduce a slightly new definition for misreports.

Definition 3.2.3 (Misreport Attack on Subjective Work Graph)**.**
Let i be an arbitrary but fixed agent with subjective work graph $G_i = (V_i, E_i, w_i)$. We say that a misreport between agents j and k has occurred if it holds for the edge weights i derived from transaction sequences TS^i_j and TS^i_k, $w^i_j(j,k) \neq w^i_k(j,k)$.

Now that we have derived a method of mapping the work that nodes have performed for one another onto a graph, we can introduce a mechanism for ranking agents by their levels of perceived cooperativeness, called accounting mechanism.

3.3. Accounting Mechanisms & Allocation Policies

The intuition behind an accounting mechanism is that it evaluates agents based on their level of cooperativeness in the network. In [26] Seuken & Parkes (2014) introduce an accounting mechanism as a function S^M which takes as input a subjective work graph from the perspective of a node i and a set of agents that request some work from that agent i. The superscript M denotes some measure or algorithm that the accounting mechanism is based on. We deviate slightly from this notation as we see no reason for the swarm of leechers to be a variable to the accounting mechanism.

Definition 3.3.1 (Accounting Mechanism)**.**
Let $i \in V$ be an arbitrary, but fixed agent in the network with subjective work graph $G_i = (V_i, E_i, w_i)$. Given some graph theoretical centrality measure M, we define an accounting mechanism as a mapping which takes as input nodes $j \in V_i$ as well as the subjective work graph of i, G_i, and returns a value denoted

$$S_i^M(G_i, j) \in \mathbb{R} \quad f.a.\ j \in V_i \setminus \{i\}.$$

Technically, the input of $S_i^M(\cdot, j)$ does not need to be the subjective work graph of i, but could be any graph G. However, in practice it only makes sense for the subjective work graph to be used. Else, the values produced will be completely irrelevant.

Here $S_i^M(G_i, j)$ determines the perceived cooperativeness of node j from the information i has gathered in the network. Every node i then obtains a set of *accounting values* for all nodes in its subjective work graph, excluding itself, which we will denote

$$S_i^M(G_i) := \left\{ S_i^M(G_i, j) \mid j \in V_i \setminus \{i\} \right\}.$$

There are infinite possibilities to define accounting mechanisms and choosing the appropriate one for a particular setting is a rather difficult task indeed. Later we will introduce a set of restrictions that accounting mechanisms must satisfy in order to be resilient against certain types of attacks and misbehaviour while simultaneously incentivising cooperativeness. Below, we introduce a set of generic examples for the reader to better understand this concept intuitively.

Example 3.3.1 (Degree-based Accounting Mechanism)**.**
As an example of a centrality measure M on a work graph G one may choose the degree centrality of nodes $j \in V$ denoted

$$deg_G(j) := \sum_{k \in V} w(k, j) - w(j, k).$$

Based on M we can derive an accounting mechanism S^M, and obtain for node i with subjective work graph G_i

$$S_i^M(G_i, j) := \sum_{k \in V_i} w_i^k(k, j) - w_i^k(j, k).$$

Note that we choose w_i^k instead of w_i^j to prevent j from successfully increasing $S_i^M(G_i, j)$ through misreports. Although this is not the point of the example.

Example 3.3.2 (BarterCast Accounting Mechanism)**.**
The BarterCast accounting mechanism is based on the maximum flow centrality measure M, which is determined by the maximum amount of data that can flow through any path connecting two nodes and can be

determined by the ford-fulkerson algorithm [12]. The BarterCast accounting mechanism is then given by

$$S_i^M(G_i, j) = \frac{arctan(maxflow(i,j) - maxflow(j,i))}{\pi/2}.$$

This is a very popular accounting mechanism which satisfies a nice property, later referred to as transitive trust. The values it assigns to nodes in the network are bounded from above by the weights of the outgoing edges from i, which is meant to limit the accounting values a node obtains from above by the amount of work this node has (indirectly) performed for i.

Example 3.3.3 (Netflow Accounting Mechanism)**.**
The Netflow (limited contribution) accounting mechanism is based on the maxflow centrality measure. An agent i determining scores of other agents in the network, will assign every node $j \in V_i$ the value

$$c_j := \max\{maxflow(i,j) - maxflow(j,i), 0\}.$$

Then i creates the new subjective work graph G_i^N, where every node j is assigned the capacity c_j then the netflow accounting mechanisms is given by

$$S_i^M(G_i, j) := maxflow_{G_i^N}(i,j).$$

An advantage of this accounting mechanism is that it's very resistant against sybil attacks, however a drawback is its lack of informativeness.

A node i holding a particular file will receive requests to share data by a set of agents in the network that are interested in the file, which we referred to earlier as a swarm of leechers. In their model, Seuken & Parkes (2014) refer to this set of agents as a *choice set* [26].

Definition 3.3.2 (Choice Set)**.**
The choice set of some node i is denoted as $C_i \subset V\backslash\{i\}$. It contains all nodes that i can seed to at a particular point in time. It can be of variable size and may even be empty depending on how many nodes happen to query node i for some contributions.

The agent now has to decide whom to contribute to based on their respective accounting values and choice set. This is done with the help of another mapping, we call *allocation policy.*

Definition 3.3.3 (Allocation Policy)**.**
Given an agent i with subjective work graph G_i, choice set C_i and a set of accounting values $S_i^M(G_i) := \{S_i^M(G_i, j) \mid j \in V_i\backslash\{i\}\}$, an allocation policy is a mapping that takes as input the set of accounting values from the perspective of i and its choice set and returns a set of agents in the choice set that i should make a contribution to. It's denoted

$$A_i : \mathbb{R}^{|V_i|-1} \times \mathscr{P}(V) \to \mathscr{P}(V)$$

with $A_i(S_i^M(G_i), C_i) \subset C_i$.

There are infinite possible different allocation policies and we will introduce a few as examples here.

Example 3.3.4 (Top n policy)**.**
Given a reputation algorithm M, subjective work graph G_i and choice set C_i of agent i the top n policy is given by

$$A_i(S_i^M(G_i), C_i) = \underset{C_i' \subset C_i \, |C_i| = n}{\arg\max} \left\{ S_i^M(G_i, j) \mid j \in C_i \right\}.$$

If there are several nodes with the same accounting values then nodes are chosen at random among these.

As a more specific case of the Top n policy, we have the winner-takes-all policy given below.

Example 3.3.5 (Winner-takes-all Policy)**.**
Given some measure M, subjective work graph G_i and choice set C_i of agent i, the winner-takes-all policy is determined by

$$A_i(S_i^M(G_i), C_i) = \arg\max \left\{ S_i^M(G_i, j) \mid j \in C_i \right\}.$$

This means i decides to perform all of its possible work for the node with the highest accounting value in the choice set. If there are several nodes who all have the same (highest) accounting values, then the "winner" is chosen at random amongst them.

A contributing node may also decide to divide its available bandwidth into equally sized chunks and to share data among several nodes in its choice set.

Example 3.3.6 (Banning Policy)**.**
Given some M, subjective work graph G_i and choice set C_i of agent i the banning policy is given by

$$A_i(S_i^M(G_i), C_i) = \left\{ j \in C_i \mid S_i^M(G_i, j) \geq \delta \right\}$$

for some arbitrary, but fixed $\delta > 0$. This means i decides to contribute to every node in its choice set whose accounting value exceeds a given lower bound.

The upper definitions can be refined in such a way that the possible contribution made by i is divided into differently sized portions which are then distributed among different agents in the choice set, whereby the contribution each agent receives is weighted by its accounting value in relation to the values of the remaining nodes. The two below are examples of such allocation policies.

Example 3.3.7 (Distribution Policy)**.**
Given some M, subjective work graph G_i and choice set C_i of agent i the distribution policy is given by

$$A_i(S_i^M(G_i), C_i) = C_i,$$

where every node $j \in C_i$ receives

$$\tilde{\omega} \cdot \frac{S_i^M(G_i, j)}{\sum\limits_{k \in C_i} S_i^M(G_i, k)}.$$

Here, $\tilde{\omega}$ is the amount of work i can perform given its bandwith limitations. In case there is only one node j in i's choice set with accounting value $S_i^M(G_i, j) = 0$, we set the amount that j received from i to $\tilde{\omega}$. If there are several nodes in C_i with $S_i^M(G_i, j) = 0$ then every node in C_i is served $\tilde{\omega} \cdot \frac{1}{|C_i|}$.

Note that this allocation policy only makes sense in the case of accounting mechanisms that only return values ≥ 0.

Example 3.3.8 (Rank-weighted Distribution Policy)**.**
Given some M, subjective work graph G_i and choice set C_i of agent i, we call

$$r_{S_i^M(G_i),C_i} : C_i \to \{1,\dots,|C_i|\}$$

the ranking, where $r_{S^M(G_i),C_i}(k)$ denotes the rank of node k in C_i, i.e. if k has the second smallest accounting value in C_i then $r_{S_i^M(G_i),C_i}(k) = 2$. If several nodes $k_1,\dots,k_n$ in C_i have the same accounting values they are assigned the same values of $r_{S_i^M(G_i),C_i}(k_i)$ f.a. $i \leq n$. The nodes following these equally ranked nodes then obtain rank $r_{S_i^M(G_i),C_i}(k_i)+1$, so we do not "skip" ranks as in the standard competition ranking.

The rank-weighted distribution policy is given by $A_i(S_i^M(G_i), C_i) = C_i$, where every node $j \in C_i$ receives

$$\tilde{\omega} \cdot \frac{r_{S_i^M(G_i),C_i}(j)}{\sum\limits_{k \in C_i} r_{S_i^M(G_i),C_i}(k)}.$$

Note that there are infinite possibilities for allocation policies and the ones above are just some intuitive examples.

Up until now our problem of incentivising cooperation through accounting mechanisms seems like a relatively easy one to solve. There are a number of different graph theoretical centrality measures that come to mind which would be suitable for accounting mechanisms to accurately capture the cooperativeness of nodes as well as many allocation policies which could effectively penalise and therefore mitigate selfish bahviour. However, additional complications arise when agents in the network begin to attack and "cheat" the system. We have already introduced the definition of a misreport attack in definition 3.1.5. However, there are a large number of other ways agents can behave maliciously making the problem much harder to solve, most notably through *sybil attacks*, which we will elaborate on later.

3.4. Misbehaviour & Attacks

Recall that it was our overarching goal to incentivise cooperative behaviour in a P2P network and therefore to prevent malicious behaviour from participants. There are many types of malicious behaviour and attacks one can perform on P2P networks. We will place special emphasis on 3 types of malicious behaviour, namely *misreporting attacks, Sybil attacks and lazy freeriding.*

3.4.1. Lazy Freeriding

The most common form of malicious behaviour is known as *lazy freeriding*, which means excessively consuming data without making proportionate contributions. So far, we have not rigorously defined what it means to be cooperative and what it means to be a lazy freerider.

Definition 3.4.1 (Lazy Freeriding)**.**
Given an agent i with respective transaction set TS_i, we say that i is a lazy freerider if the aggregated amount of data they have consumed is much larger than the amount they have contributed, i.e. for some fixed $c \leq 0$ it holds

$$\sum_{j \in V} \sum_{n \in \mathbb{N}_{\leq T_i}} t_i(n, j) \leq c.$$

We can rewrite this in terms of the work graph as

$$\sum_{j \in V} w(j,i) - w(i,j) \leq c.$$

Alternatively, we can label a node i a lazy freerider if the ratio of contribution to consumption exceeds some arbitrary but fixed lower bound $c' \leq 1$.

$$\frac{\sum_{j \in V} \sum_{n \in \mathbb{N}} \max\left\{t_i(n,j), 0\right\}}{\sum_{j \in V} \sum_{n \in \mathbb{N}} -\min\left\{t_i(n,j), 0\right\}} \leq c',$$

or written in terms of edge weights in the work graph

$$\frac{\sum_{j \in V} w(j,i)}{\sum_{j \in V} w(i,j)} \leq c'.$$

Each of these two definitions captures the concept of lazy freeriding from a slightly different angle. We prefer the latter definition, seeing as we find the proportion of up-to downloads more appropriate than strictly the difference between the two. This is because we think the difference should be allowed to be bigger as the absolute values of the two grow. In later experiments we will stick the former though, as we will limit the number of interactions nodes can have. In that case the former definition of lazy freeriding becomes more informative.

This is the problem accounting mechanisms were introduced to prevent. Accounting mechanisms are meant to prevent lazy freeriding and facilitate cooperation by punishing selfish behaviour and rewarding altruistic behaviour. This is done by assigning nodes that contribute more and consume less than other nodes, higher accounting values, such that they are more likely to receive work later on. Below, we introduce a sufficient, but not necessary requirement for an accounting mechanism to prevent lazy freeriding.

Definition 3.4.2 (Positive-Report Responsiveness)**.**
Given a subjective work graph $G_i = (V_i, E_i, w_i)$ derived from a subjective transaction set TS^i and agent j with transaction set TS^i_j of length T^i_j. If i learns of another transaction, j has participated in

$$t^i_{j,T^i_j+1} = (j,k,r),$$

with $r > 0$. Then i updates their transaction set to obtain

$$TS'^i_j = TS^i_j \cup \left\{t^i_{j,T^i_j+1}\right\}$$

and derives a new subjective work graph $G'_i = (V'_i, E'_i, w'_i)$ with the updated edge

$$w'^i_j(k,j) = w^i_j(k,j) + r.$$

Then it must hold

$$S^M_i(G'_i, j) \geq S^M_i(G_i, j)$$

and

$$S^M_i(G'_i, k) \leq S^M_i(G_i, k).$$

More rigorously, we define an accounting mechanism to be *strictly positive-report responsive* if there exists some $\varepsilon > 0$ such that it holds under the exact same conditions above for any transaction of weight $\geq r$ or any sequence of transactions (between the same parties) of aggregated weight $\geq r$:

$$S^M_i(G'_i, j) - S^M_i(G_i, j) \geq \varepsilon$$

and

$$S^M_i(G'_i, k) - S^M_i(G_i, k) \leq -\varepsilon.$$

As an example of a combination of accounting mechanism and allocation policy that prevents lazy freeriding very effectively we look at the banning policy in combination with an accounting mechanism that satisfies strict positive-report responsiveness.

Example 3.4.1.
Let $i \in V$ be a lazy freerider and let all agents in the network adopt the banning policy together with an accounting mechanism S^M that satisfies strict positive-report responsiveness for some $\varepsilon > 0$. Lastly, assume i's transaction sequence is reported to all other nodes in the network without any misreports. Then there exists a fixed $c' \in \mathbb{R}$ such that regardless of which nodes i queries and how many contributions i makes to others, it will always hold

$$\lim_{T_i \to \infty} \sum_{j \in V} \sum_{n \in \mathbb{N}_{\le T_i}} t_i(n, j) \ge c'.$$

Proof. The main idea behind this is that if i leeches continuously from the network, due to the strict positive-report responsiveness and the assumed misreport-proofness its accounting values will go beneath δ after a finite number of transactions, from the perspective of all honest nodes in the network. At this point i will no longer have a positive probability of being served by another node due to the banning policy of all other honest nodes.

□

The upper example may make it seem like the banning policy is a very good allocation policy for a P2P network, but it actually has a major drawback, namely the fact that it acts as a bottleneck for the distribution of data. This is because agents will stop serving one another under certain conditions. However, bandwidth cannot be stockpiled and hence there is no reason for a node not to make contributions to other nodes in the network. The point behind an allocation policy is that it's supposed to choose the nodes in the choice set that have priority and not exclude nodes entirely. Hence, despite it very effectively preventing lazy freeriding it is not an ideal allocation policy.

3.4.2. Misreports

We have already defined misreport attacks in definitions 3.1.5 and 3.2.3 and have seen in example 3.3.1 the effects it can have on the values accounting mechanisms return. But so far, we have not yet discussed how to prevent them or at least how to render them ineffective.

In [24] Seuken & Parkes (2010) introduce the definition of misreport-proof as follows.

Definition 3.4.3 (Misreport-Proofness on the Choice Set)**.**
An accounting mechanism S_i^M of agent i with subjective work graph G_i and choice set C_i is misreport-proof if for any agent $j \in C_i$ that commits a misreport attack, leading to the subjective work graph G_i' it holds

$$\begin{aligned} &S_i^M(G_i', j) \le S_i^M(G_i, j) \\ &S_i^M(G_i', k) \ge S_i^M(G_i, k) \quad \text{f.a. } k \in C_i \backslash \{j\}. \end{aligned}$$

This particular definition of misreport-proofness was introduced with a mechanism called DropEdge in mind. We will introduce this mechanism below. However, with the TrustChain datastructure in mind, this definition can be strengthened to misreport-proofness on the entire work graph.

Definition 3.4.4 (Misreport-Proofness)**.**
An accounting mechanism S_i^M of agent i with subjective work graph G_i is misreport-proof if for any agent $j \in V_i$ that commits a misreport attack, leading to the subjective work graph G_i' it holds

$$\begin{aligned} &S_i^M(G_i', j) = S_i^M(G_i, j) \\ &S_i^M(G_i', k) = S_i^M(G_i, k) \quad \text{f.a. } k \in V_i \backslash \{j\}. \end{aligned}$$

In [26] Seuken & Parkes (2014) introduce a mechanism called *Drop-Edge*. Notation-wise we deviate slightly from their definition while maintaining the same concept.

Definition 3.4.5 (Drop-Edge Mechanism)**.**
Given agent i with subjective work graph G_i the Drop-Edge mechanism is given by a mapping D from the space of subjective work graphs into itself, such that

$$D(G_i, C_i) := G_i^D$$

with edge weights w_D^i satisfying

$$\begin{aligned}
&\forall (j,k) | i \in \{j,k\} : w_D^i(j,k) = w_i^i(j,k)\\
&\forall (j,k) | j,k \in C_i : w_D^i = 0\\
&\forall (j,k) | j \in C_i \, k \not\in C_i : w_D^i(j,k) = w_k^i(j,k)\\
&\forall (j,k) | k \in C_i \, j \not\in C_i : w_D^i(j,k) = w_k^i(j,k)\\
&\forall (j,k) | j,k \not\in C_i, i \not\in \{j,k\} : w_D^i(j,k) = \max\left\{ w_j^i(j,k), w_k^i(j,k) \right\}
\end{aligned}$$

Missing values in the max operator are set to 0.

They proved that this mechanism successfully disincentivises misreporting by eliminating any reward a misreport will have for the node that commits the misreport. Consequently, we find that for any accounting mechanism S_i^M, $S_i^M \circ D$ is misreport-proof on the choice set and i obtains a subjective work graph with single edge weights $w_D^i(j,k)$ for any $j,k \in V_i$.

Remark 3.4.1. Note that this mechanism is **only** misreport-proof on the choice set and not misreport-proof in the sense of definition 3.4.4. It's also only resistant to misreport attacks on the subjective work graph, and not misreport attacks in the sense of definition 3.1.5. Another point to mention here is that while an agent can never benefit from their own misreport they may be able to benefit from another agent's misreport. Recall that we mentioned in remark 3.2.3 that some agents may misreport in such a way that the edge weights in the subjective work graph remain consistent. In such a case it is in both parties' best interest to misreport about a transaction and both parties benefit from this misreport. DropEdge does not prevent this type of misreporting. In DropEdge this means that even if a node reports honestly and its transaction partner misreports it may obtain higher accounting values than it would have if both agents had reported honestly. One might at first think that this is unlikely to occur as it is not in the interest of both participants, but the example given below shows that this is not actually the case.

Example 3.4.2.
Let i be an honest agent with accounting mechanism S_i^M where M is given by the personalised PageRank as given by Stannat et al. (2019) [28]. Now let G_i be the subjective work graph of agent i as seen in figure 3.5 below.

We see that in the figure on the right j has committed a misreport, namely it has reported (a) transaction(s) of weight 1 from j to k. If k is honest then k reports an edge weight $w(j,k) = 0$. Now, if k is in the choice set of i and j is not and if i applies the DropEdge mechanism to its accounting mechanism S_i^M then k is rewarded by j's misreport.

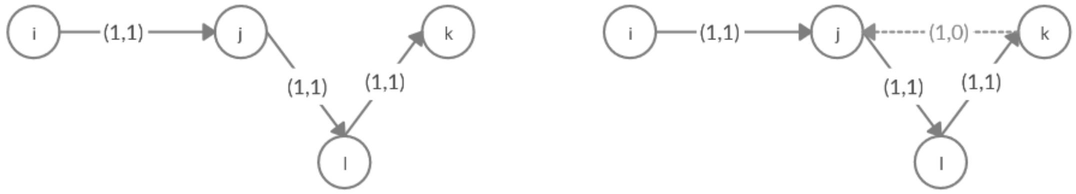


Figure 3.5: Misreport on PageRank and Drop-Edge

If both agents k and j were to misreport on this edge weight such that $w_i(j,k) = (1,1)$ then k would also "get away" wit this misreport, despite the DropEdge mechanism, so long as j is not in i's choice set. This is proof that the DropEdge mechanism isn't misreport-proof in the sense of definition 3.4.4 and that it is only resistant to misreports on the subjective work graph according to definition 3.2.3.

In order to achieve general misreport-proofness, we have a stronger mechanism which we introduced earlier in chapter 1 as TrustChain [20]. We will now formalise TrustChain mathematically as a way of enhancing transactions to render misreports detectable. The concept of TrustChain entails two definitions which we will not introduce here in any detail, namely those of hash functions and digital signatures. We assume the reader to be familiar with these basic cryptographic concepts and refer to Smart et al. (2016) for the details of these concepts [27].

Definition 3.4.6 (TrustChain)**.**
Let j, k be two arbitrary agents in the network. As in definition 3.1.1 we write a transaction t as a tuple containing the two participants and the amount of data transferred, but add a set of additional values to it. A transaction in the TrustChain datastructure from j to k, of weight w is then denoted $\tilde{t}$ and given by

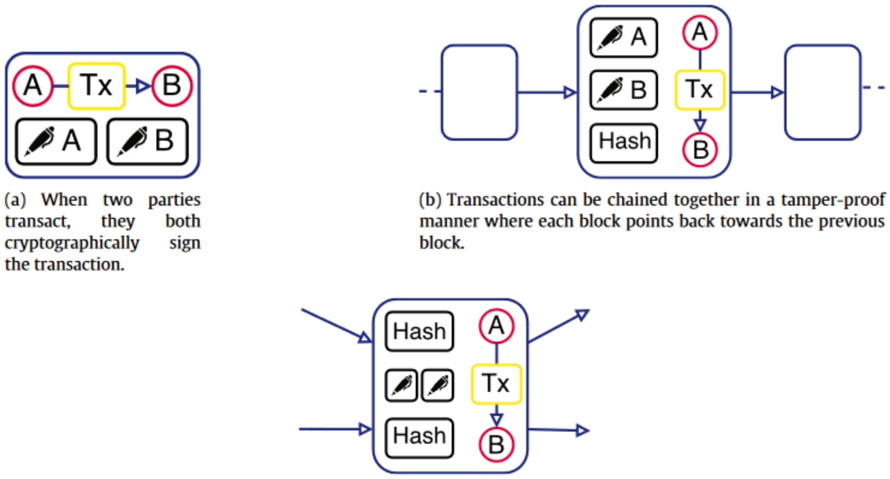
$$(j, k, w, id, hash_j, hash_k, sig_j, sig_k).$$

The value id is the unique identifier of the transaction such that no two transactions between the same nodes can be confused. The values $hash_j$ and $hash_k$ are hash pointers to the transactions that precede the given transaction in the transaction sequences of both participants. I.e. if $(j, k, 5, id, hash_j, hash_k, sig_j, sig_k)$ corresponds to $\tilde{t}_{j,n}$ and $\tilde{t}_{k,m}$ then $hash_j$ and $hash_k$ are given by a hash function h applied to $\tilde{t}_{j,n-1}$ and $\tilde{t}_{k,m-1}$. If $n = 1$ or $m = 1$ then we set $hash_j = 0$ or $hash_k = 0$. Finally sig_j and sig_k are the digital signatures of j and k.

Consequently, we write the transaction sequences as $\tilde{TS}_i$, the transaction functions as $\tilde{t}_i$ and the transaction set as $\tilde{TS}$. Nodes share their transaction sequences $\tilde{TS}_i$ with one another just like before and every agent i then obtains a subjective transaction set $\tilde{TS}^i$ as before.

Now i can derive the subjective work graph from its subjective transaction set $\tilde{TS}^i$ analogously to definition 3.2.1 with the help of a mapping function g and obtain $\tilde{G}_i = g(\tilde{TS})$.

Example 3.4.3.
As a visualisation of a set of transactions in the TrustChain data structure we see the images in figures 3.6 and 3.7 given below



(a) When two parties transact, they both cryptographically sign the transaction.

(b) Transactions can be chained together in a tamper-proof manner where each block points back towards the previous block.

(c) To increase the resistance against tampering, each block also references a block in the chain of the counterparty. This ensures that each block has two incoming and two outgoing pointers.

Figure 3.6: TrustChain Transaction Structure [20]

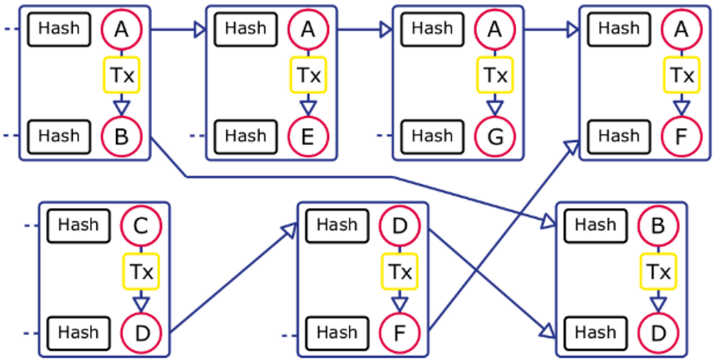


Figure 3.7: TrustChain Hash Pointers [20]

Theorem 3.4.1.
Given an adequate transaction reporting scheme TrustChain makes any misreport detectable in finite time.

Proof. The proof to this has been given by Harms et al. (2018), in which a History-Exchange policy was introduced which we will not elaborate on. [8]. □

Theorem 3.4.2.
Any positive-report responsive accounting mechanism S^M on a subjective work graph $\tilde{G}_i$, derived from the TrustChain based subjective transaction set $\tilde{TS}^i$ is misreport-proof in accordance with defintion 3.4.4.

Proof. Let i be the node with subjective transaction set $\tilde{TS}^i$ and let j be a malicious agent attempting to misreport to i. There are 4 ways j can go about this.

(i) j drops a transaction $\tilde{t}_{j,n}$ $(n < \tilde{T}_j)$ from its transaction sequence $\tilde{TS}_j$.

(ii) j drops transaction $\tilde{t}_{j,\tilde{T}_j}$ from its transaction sequence $\tilde{TS}_j$.

(iii) j adds a transaction $\tilde{t}_{j,\tilde{T}_{j+1}}$ to the end of its transaction sequence.

(iv) j adds a transaction $\tilde{t}$ into its transaction sequence, but not at the end.

We will prove that all 4 of these types of attacks are prevented, or at least exposed by the TrustChain mechanism.

(i) If j drops transaction $\tilde{t}_{j,n}$ from its transaction sequence and reports the altered sequence to i, i will be able to detect the misreport, by looking at the hash pointer in $\tilde{t}_{j,n+1}$ and comparing it to the hash value generated by $\tilde{t}_{j,n-1}$. i will then notice that these hash values don't add up and will conclude that j has committed a misreport.

(ii) Assume j drops transaction $\tilde{t}_{j,\tilde{T}_j}$ from its transaction sequence $\tilde{TS}_j$ and the other participant k keeps this transaction in their sequence $\tilde{TS}_k$. Then once i has queried k's transaction sequence, i will notice the misreport. Because S^M is positive-report responsive, we know that for one of the two agents there is no incentive to drop this transaction and therefore we know that one of the two will always keep it, so long as there is no collusion.

i will now receive the reports $\tilde{TS}_k^i$ and $\tilde{TS}_j^i$ where the former contains the dropped transaction $\tilde{t}$, while the latter does not. Due to the fact that $\tilde{t}$ contains the hashes of transactions in j's and k's transaction sequences, i will know of the misreport and will be able to attribute it to j, due to the digital signatures in the transaction.

(iii) If j adds a transaction to its transaction sequence $\tilde{TS}_j$ that has not actually occurred, we face the same situation as in point (ii), where there is one transaction sequence with a missing final block. Just as before i will be able to determine this, due to the same reason as discussed above. And using the digital signature scheme i can determine that j was responsible for the misreport and not k.

(iv) Lastly, if j attempts to fraudulently add a transaction $\tilde{t}$ into its transaction sequence $\tilde{TS}_j$, let's say in between transactions $\tilde{t}_{j,m}$ and $\tilde{t}_{j,m+1}$ then looking at the hash values in $\tilde{t}_{j,m+1}$, i will see that they reference $\tilde{t}_{j,m}$ and not $\tilde{t}$. Seeing as looking at the hash values in $\tilde{t}_{j,m+2}$ returns transaction $\tilde{t}_{j,m+1}$ i can see that the misreported transaction was $\tilde{t}$. The misreport has been identified.

We now write $TC(\tilde{G}_i) := G_i^{TC}$, with edge weights w_{TC}^i that are given by a single value as opposed to a tuple through

$$\forall (j,k) \mid i \in \{j,k\} : w_{TC}^i(j,k) = w_i^i(j,k)$$
$$\forall (j,k) \mid i \not\in \{j,k\}, w_j^i(j,k) \neq w_k^i(j,k) : w_{TC}^i(j,k) = \max\left\{w_j^i(j,k), w_k^i(j,k)\right\}.$$

If one of the two values is zero, because one of the two nodes has not shared their chain yet, then we set the weight of the edge to 0. Note that the trick of taking the max only works because the only misreport that may not be detected right away is that of dropping a transaction from one's chain. This only works in subjective work graphs with double edges.

Hence, we find that $S_i^M \circ TC$ is misreport-proof in the sense of definition 3.2.3 for any accounting mechanism S_i^M that satisfies positive-report responsiveness. □

Note that in theorem 3.4.2 above, we can even weaken the assumption of positive-report responsiveness to one where any transaction report between two agents i and j will increase the accounting value of at least one of the two. This way any form of block-withholding attack is disincentivised. This holds for almost all reasonable accounting mechanisms and the only way a block-withholding attack could go unnoticed is if both transaction parties lose some accounting values in response to the transaction. Any reasonable accounting mechanism would not satisfy this however.

Knowing that the TrustChain mechanism renders any accounting mechanism misreport-proof, we assume in all further analysis of accounting mechanisms that the transaction set has been built up on the TrustChain structure. Hence, from here on out we will no longer analyse misreport-proofness and solely focus on preventing freeriding and sybil attacks.

3.4.3. Sybil Attacks

We have already introduced the concept of sybil attacks in which agents create multiple fake accounts which report counterfeit transactions amongst one another to the network, in chapter 1. We will formalise this type of attack mathematically below. Note that the mechanisms we introduced in subsection 3.4.2 cannot prevent this type of attack as it is fundamentally different from a misreport attack.

DropEdge cannot prevent this as both parties involved in the fake transaction have added it to their transaction sequence and hence it holds for two sybil nodes s_1, s_2 and any honest node i, $w_i^{s_1}(s_1, s_2) = w_i^{s_2}(s_1, s_2)$. TrustChain cannot prevent this type of attack either as both parties involved in the fake transaction give their digital signatures on it and include it in their transaction sequence in accordance with the hash pointers.

Definition 3.4.7 (Sybil Attack)**.**
Given an objective work graph $G = (V, E, w)$, a sybil attack by agent $j \in V$ is given by a set of n new identities $S = \{s_{j_1}, \dots, s_{j_n}\}$ and a set of edges $E_S \subset S \cup \{j\} \times S \cup \{j\}$ with edge weights $w_S : S \cup \{j\} \times S \cup \{j\} \to \mathbb{R}$. Additionally, there must be a set of attack edges E_{attack}, i.e. $w_{attack} : V \backslash \{j\} \times S \cup \{j\} \to \mathbb{R}_{\geq 0}$. We label a sybil attack by node $j \in V$ with n fake identites σ_j^n.

The new work graph after the sybil attack is given by $G' := G \downarrow \sigma_j^n = (V', E', w') = (V \cup S, E \cup E_S \cup E_{attack}, w')$, where

$$w'(u, v) = \begin{cases} w(u, v), & \text{if} \quad u, v \in V \backslash \{j\} \\ w_S(u, v), & \text{if} \quad u, v \in S \cup \{j\} \\ w_{attack}(u, v), & \text{if} \quad u \in V \backslash \{j\}, v \in S \cup \{j\} \end{cases}.$$

We define a sybil attack on a subjective work graph equivalently by $G_i' := G_i \downarrow \sigma_j^n$.

Attack edges correspond to real transactions in which the attacker makes a legitimate donation to some nodes in the honest part of the network from one or more of the nodes they create. These are the basis of every sybil attack and in a network in which the accounting mechanism satisfies a property called path-responsiveness, they are a requirement for the attack to have any effect. In the case of an accounting mechanism that does not satisfy this property they may also be dropped.

Definition 3.4.8 (Path-Responsiveness)**.**
Given a subjective work graph $G_i = (V_i, E_i, w_i)$ we say that an accounting mechanism S^M satisfies path-responsiveness if it holds

$$S_i^M(G_i, k) > 0 \Rightarrow \exists j_1, \dots, j_n \in V_i : w_i(i, j_1), w_i(j_1, j_2), \dots, w_i(j_{n-1}, j_n), w_i(j_n, k) > 0.$$

This means that in order for a node k to have a positive value in the accounting mechanism from the perspective of another node i there needs to exist at least one path connecting i and k. This path needs to correspond to work indirectly performed by k for i through the other nodes in the path.

Path-responsiveness is a rather important definition in the context of sybil attacks and sybil-resistance of accounting mechanisms. It is useful as it prevents agents from obtaining accounting values ≥ 0 without performing at least some honest work.

The edges in E_S are the edges connecting agents in the sybil region to one another. These are the "fake edges" which represent work that hasn't actually been performed. The point behind them is to amplify the reputation agents in the sybil region have honestly obtained through the attack edges.

Note that it is common for the sybil region to be densely connected and containing many nodes, forming a separated cluster in the network with relatively few edges connecting it to the honest region of the network, seeing as these types of edges are given by actual work, which is costly. The obvious way to detect such a type of attack would be through community detection algorithms such as the *minimum-cut method* that detect densely connected regions in the work graph [14]. However, sybil attacks can take many different forms and shapes and therefore this is not a feasible solution for *all* types of sybil attacks. Depending on which accounting mechanism and allocation policy are used, sybil attacks may look very different than others. As we can see in examples 3.4.4 and 3.4.5.

Example 3.4.4.

Let G_i be an arbitrary subjective work graph of agent i with the degree-based accounting mechanism S_i^{deg} given in example 3.3.1. A typical sybil attack on this accounting mechanism by malicious agent j would be given by j creating n sybils that all connect to j and reporting these edges to i, i.e. $w_i(s_{jk}, j) = c$ f.a. $k = 1, \ldots, n$. A visualisation of this can be found in figure 3.8 below. In the case of the degree-based accounting mechanism there is no need for attack edges as S_i^{deg} does not satisfy path-responsiveness.

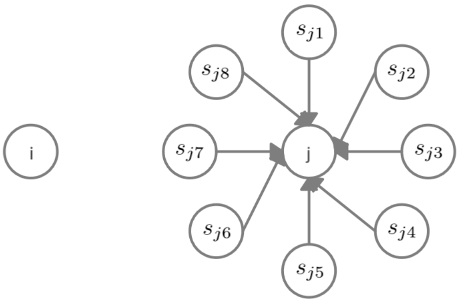


Figure 3.8: Sybil Attack on Degree-based Accounting Mechanism

Example 3.4.5.

Let G_i be an arbitrary subjective work graph of agent i with the Maxflow accounting mechanism S_i^M given in example 3.3.2. A typical sybil attack by agent j on this accounting mechanism would be given by j creating n sybil identities that all perform some counterfeit work for j as visualised in figure 3.9 below.

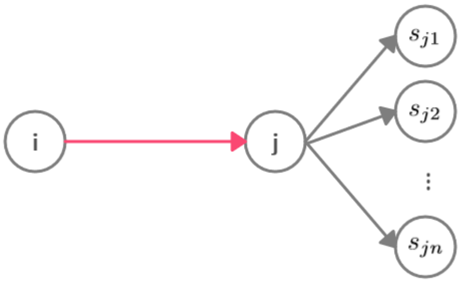


Figure 3.9: Sybil Attack on Maxflow Accounting Mechanism

In [23] Seuken & Parkes (2011) differentiate between *active* and *passive* sybil attacks. In a passive sybil attack, attack edges are only connected to one and the same node, i.e. $w'(k,s) = 0$ f.a. $s \in S, k \in V \backslash \{j\}$. In an active sybil attack every node in the sybil region may be connected to the honest region of the network, as visualised in figure 3.10 below. In [19] by Otte et al. (2016) a sybil attack is defined such that it is perpetrated by a set of nodes $J \subset V$. This was done to combine both definitions of active and passive sybil attacks in one. We find this definition slightly ineffective as it seems to combine sybil attacks with collusion attacks, in one definition. Collusion attacks occur when several independent actors collude to achieve a common goal, which are obviously different from a single agent creating multiple fake identities. Hence, our definition above deviates from the existing ones a bit.

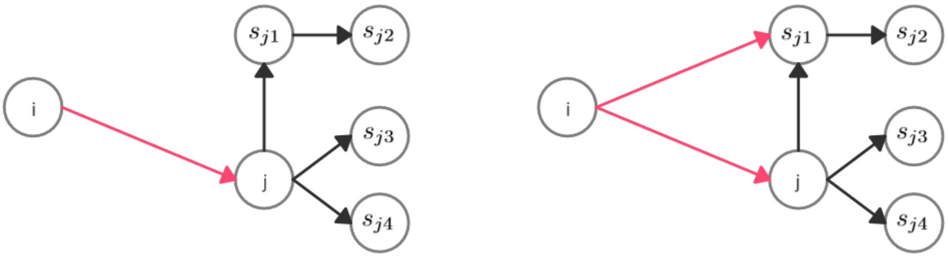


Figure 3.10: Passive vs Active Sybil Attacks

Seeing as in the active sybil attack defined above, the work graph does not reveal who the sybil attacker is and who their fake identities are (this is only the case in [19] and in [23]), we can simply drop the j from σ_j^n. In the following chapters we will analyse this type of attack and its effects in much more detail.

We conclude this section by stating that one aims to devise accounting mechanisms that are resistant to such types of attacks. By this we mean that an accounting mechanism should by design dampen the effect that fake identities and fake accounts have on the increase in accounting values and the consequent increase in work they can consume. Ideally, they should not increase at all, however this is rather difficult to achieve. In the following chapter we will further analyse the effects of sybil attacks and their gain for the perpetrating node.

4 Mathematical Framework for Sybil Attack Gain

The point behind a sybil attack was to artificially increase one's accounting values in an attempt to subvert the reputation mechanism and obtain more data than one is actually entitled to. We then say that a Sybil attack is beneficial if the attacking agent or one of its sybils is chosen to receive some work when without the attack it would not have been. There's 4 ways of how this may be the case.

Definition 4.0.1 (Beneficial Attacks)**.** A sybil attack by agent j is considered beneficial if for some agent i with choice set C_i, subjective work graph G_i, accounting mechanism S_i^M and allocation policy A_i that would pick some agents $A_i(S_i^M(G_i), C_i) \subset C_i$, we obtain one of 4 outcomes

- j's score is increased such that $j \in A_i(S^M(G_i'), C_i')$ when before $j \not\in A_i(S^M(G_i), C_i)$
- Other agents' scores are lowered such that $j \in A_i(S^M(G_i'), C_i')$ when before $j \not\in A_i(S^M(G_i), C_i)$
- A sybil s is assigned a score such that $s \in A_i(S^M(G_i'), C_i')$ when before $s \not\in A_i(S^M(G_i), C_i)$
- Other agents' scores are lowered such that $s \in A_i(S^M(G_i'), C_i')$ for some sybil s when before $s \not\in A_i(S^M(G_i), C_i)$.

However, the upper conditions may be satisfied for a single attacking node, or for several. They may also hold true for multiple nodes i from which the attacking agents could leech. Additionally, one or more of them may also still be satisfied after one or more of the attackers have received some work.

Hence, we see that some sybil attacks may be more beneficial than others depending on how much more work the attacker can consume than they were actually entitled to before the attack occurred. The upper definition is therefore rather inaccurate as it does not capture *how* beneficial an attack is.

4.1. Determining Cost & Profit of Sybil Attacks

We now want to introduce an exact definition of *how* beneficial a sybil attack is, which will be determined by the ratio of the work invested into the attack and the amount of work that the attacker(s) can gain through it. Seuken & Parkes (2014) introduce the following definition of Sybil attack profit.

Definition 4.1.1 (Sybil Attack Benefit).
Let j be a malicious node perpetrating a sybil attack σ_j^n on the work graph G, resulting in work graph G'. Here n is variable. Now let $\omega_-^n > 0$ denote the amount of work j has invested into the sybil attack σ_j^n and let ω_+^n be the amount of work that j and its sybils can consume after the attack has been carried out as a result of it. Then σ_j^n is called

Strongly Beneficial if $\omega_+^n > 0$ and $\omega_-^n = 0$ or if $\lim_{n\to\infty} \frac{\omega_+^n}{\omega_-^n} = \infty$,

Weakly Beneficial if $\omega_+^n > 0$ and $\omega_-^n > 0$ and $\exists c > 0 : \lim_{n\to\infty} \frac{\omega_+^n}{\omega_-^n} \geq c$.

It's almost impossible to prevent weakly beneficial sybil attacks from happening. Even though they might seem problematic they actually, if scaled, require an attacker to invest infinite resources in order to obtain infinite resources. A strongly beneficial sybil attack is much more fatal, seeing as an attacker can leech infinite resources without contributing a proportionate amount. This can bring an entire network to a stand-still and is therefore much more important to prevent than weakly beneficial sybil attacks. While it may be nice to find some upper bound for the value of c in the definition above, it will be our goal to prevent strongly beneficial sybil attacks.

While the values for ω_-^n and ω_+^n may be intuitively clear, we realise that the more one thinks about them, the more involved these definitions actually turn out to be. In the existing literature, they have been introduced as above without any further explanation. In this chapter we aim to refine this definition and determine more rigorous definitions for the cost and profit of sybil attacks. For the value ω_-^n this is not so difficult, while ω_+^n is much less clear. We begin by introducing the cost or investment of a sybil attack.

Definition 4.1.2 (Sybil Attack Cost).
Given an objective and a subjective work graph $G := (V, E, w)$, $G_i := (V_i, E_i, w_i)$, let σ_j^n be a sybil attack of size $n \in \mathbb{N}$ with Sybil region $S = \{s_{j1}, \ldots, s_{jn}\}$, whereby n is not fixed. Take $G' := (V', E', w')$ and $G_i' := (V_i', E_i', w_i')$ as defined above. We define ω_-^n as the amount of work invested into the sybil attack. This is the aggregated amount of work that the attacker and its sybil nodes have performed for the network, given by the collective weight of all incoming edges from the honest region.

$$\omega_-^n = \sum_{u\in V\setminus\{j\}} \sum_{v\in S\cup\{j\}} w(u, v).$$

Note that at the moment of the sybil attack the newly created nodes, i.e. the sybil nodes have not received any work from other nodes in the network yet. This means there are no outgoing edges from nodes in S into the honest part of the network. This, of course, does not hold for j itself as it may have already participated in the network before launching the sybil attack. In the case of a passive sybil attack it holds $w(u, v) = 0$ f.a. $u \in V\setminus\{j\}$, $v \in S$, as all attack edges are connected to j itself. In this case we obtain

$$\omega_-^n = \sum_{u\in V\setminus\{j\}} w(u, j).$$

This is the amount of actual work the attacker has to invest into the attack, in order to boost its own and its sybils reputation, relative to the rest of the network, and consequently obtain work from the rest of the network. The edges in the work graph corresponding to this work performed are what we called *attack edges.* Note that these edges are indispensable for a sybil attack. If no such edges exist then no node in the sybil region, including j can increase their accounting values from the perspective of any node outside of the sybil region. At least, as long as our requirement of path-responsiveness is satisfied.

Inversely, we should define the reward/profit ω_+^n of a sybil attack by the aggregated amount of work, all nodes in the sybil region can collectively consume after the attack has been carried out. If the attack enables the sybils to consume much more data than they collectively performed, the attack will be considered detrimental to the network and beneficial for the attacker. Based on our definition of lazy freeriding in chapter 2, we will want to determine $\frac{\omega_+^n}{\omega_-^n}$.

However, computing the value of ω_+^n turns out to be a rather difficult task. There are a number of different factors this value depends on, such as the accounting mechanism and the allocation policy of agents in the network as well as the state of the interaction graph. More importantly, it depends on the dynamic of the interaction graph as time progresses and interactions between different agents occur. Other agents participating in transactions, even if these don't involve the attackers themselves, will affect the reputation values of the attackers and therefore also the value of ω_+^n.

In particular, this value turns out to be probabilistic in nature. A node j that queries another node i for some data will be served if at the given point in time it happens to be the node with the right accounting value $S_i^M(G_i, j)$ to be chosen by i's allocation policy. This, of course, does not only depend on j's accounting value, but on who else happens to query i at this given point, which j has no control over. In order for us to be able to gauge this value, we introduce an interaction model among agents, with the intent to approximate the dynamics of real-world P2P networks. In fact, we assume a model in which the choice set will be random, following a given distribution, explained below.

4.1.1. Interaction Model

We say that the network operates in rounds, for simplicity. In each round, every honest node k will with a given probability q, $(Ber(q))$ query some other randomly chosen node i in the network. The node that is queried is chosen following the uniform distribution with probability $\frac{1}{|V\setminus\{k\}|}$. Every honest node that has been queried will respond by doing some work for the node(s) in its choice set, chosen by its allocation policy. For simplicity, the amount of work will be a fixed value $\tilde{w}$. Let C_i be the set of all agents, requesting some work from agent i in a given round. Then it follows $|C_i| \sim \mathscr{B}in(|V|-1, \frac{q}{|V|-1})$. Due to independence of the two random variables (uniform and bernoulli) we can multiply the probabilities and obtain the binomial distribution. An interesting property of this model is that for $|V| \to \infty$ the binomial distribution $\mathscr{B}in(|V|-1, \frac{q}{|V|-1}) \to Poi(q)$ converges to the poisson distribution as the network size goes to infinity. The last necessary assumption we make for this model is that $|V| = \infty$. The reason we make this assumption will become apparent later, in definition 4.1.3

We assume that sybil attackers do not have full knowledge on the state of the interaction graph and that targeted attacks are therefore impossible. Instead, we assume that sybil attackers have no better option than to leech from randomly chosen nodes in the network. Attackers have no option of being strategic in their attack as they do not know the subjective work graph of any of the honest nodes they may want to target and can therefore not gauge the accounting values a possible victim assigns to other nodes in the network. We realise that this is a rather restrictive assumption and it may be more true for some accounting mechanisms than others. But we feel that it is a necessary one to make for any generic model. Additionally, we assume that a sybil attacker cannot attack several agents with the same node, i.e. a single sybil can only ever find itself in the choice set of a single honest node. Lastly, all honest nodes in the network are assumed to share the same accounting mechanism and allocation policy.

There are some inevitable inaccuracies in this model. In real file-sharing networks nodes query other nodes, based on the files they are interested in. To assume that nodes are chosen with equal probability is not 100% realistic, as there may be nodes holding more sought-after files than others. Some agents may not hold any files at all and will therefore not receive any queries at all. We also disregard the possibility of nodes going offline and therefore not responding to queries and/or not querying other nodes in the network. A fixed size $\tilde{w}$ of every transaction is also somewhat unrealistic as files vary in size, but we see it as a necessary restriction for the model. Lastly, the notion that the network operates in rounds is also somewhat contrived as real-world networks are continuous and requests for files do not come in rounds.

4.1.2. Choosing an Allocation Policy

Our model should be agnostic of accounting mechanisms, as these are the subject of our research, but in order to gauge the profit we will assume that all participating honest nodes will adopt the winner-takes-all allocation policy. The reason we believe this is a good choice is that of all policies mentioned in examples 3.3.4 to 3.3.8 it is the most resistant to large-scale sybil attacks, given an unspecified accounting mechanism. The justification of this claim is rather complex and will be explained in the following propositions and remarks below.

The first point we make is that the expected gain of a sybil attack is higher for the top n policy than for the winner-takes-all allocation policy. By this we mean the expected value of the amount of data a sybil attacker can consume in a single round. In particular, it will follow that the largest expected profit a sybil attacker can gain in a single round is also smaller for the winner-takes-all allocation policy.

Proposition 4.1.1.

Let $j \in V$ be a sybil attacker with sybil region S. Let the choice sets of nodes in the network be assembled according to the protocol discussed in subsection 4.1.1. Then the top n allocation policy for some arbitrary, but fixed $n \in \mathbb{N}_{\geq 1}$ will yield a higher expected profit for the sybil attacker than the winner-takes-all allocation policy, in a single round.

Proof. Let $i \in V$ be some arbitrary but fixed node and let C_i be a randomly put-together choice set of i. Now let $S' \subset S \cup \{j\}$ be some set of nodes that attack node i. It then follows

$$\mathbb{P}\left(\operatorname*{arg\,max}_{k \in C_i}\left(S_i^M(G_i,k)\right) \in S'\right) \leq \mathbb{P}\left(\operatorname*{arg\,max}_{C_i' \subset C_i, |C_i|=n}\left\{S_i^M(G_i,k) \mid k \in C_i\right\} \cap S' \neq \varnothing\right).$$

Hence the probability of the attacker obtaining $\tilde{\omega}$ units of work is lower in the case of the winner-takes-all allocation policy.

This is obvious, as the probability of a sybil node being the highest-ranking in C_i is lower than the probability of one or more sybil nodes belonging to the highest-ranking n nodes, when the choice set is generated as given by the model above. One is trivially a subset of the other.

It therefore follows for any partition of S into subsets $S' := \left\{S_i \subset S \mid i \in V \backslash \{j\}\right\}$, each of which attacks a different honest agent i that it holds

$$\mathbb{P}\left(\bigcup_{i \in V \backslash \{j\}}\left\{\operatorname*{arg\,max}_{k \in C_i}\left(S_i^M(G_i,k)\right) \in S_i\right\}\right) \leq \mathbb{P}\left(\bigcup_{i \in V \backslash \{j\}}\left\{\operatorname*{arg\,max}_{C_i' \subset C_i, |C_i|=n}\left\{S_i^M(G_i,k) \mid k \in C_i\right\} \cap S_i \neq \varnothing\right\}\right).$$

Now it follows for the expected amount of work attacker j can expect to obtain with a given partition S'

$$\sum_{i \in V \backslash \{j\}} \tilde{\omega} \cdot \mathbb{P}\left(\operatorname*{arg\,max}_{k \in C_i}\left(S_i^M(G_i,k)\right) \in S_i\right) \leq \sum_{i \in V \backslash \{j\}} \tilde{\omega} \cdot \mathbb{P}\left(\operatorname*{arg\,max}_{C_i' \subset C_i, |C_i|=n}\left\{S_i^M(G_i,k) \mid k \in C_i\right\} \cap S_i \neq \varnothing\right).$$

□

Remark 4.1.1.

For both of these allocation policies it is most strategic for the attacker to distribute their sybil nodes' queries over the entire network and not to query the same node with several of its sybil identities. In case of the winner-takes-all policy this is obvious as there cannot be two nodes that are both the highest ranking nodes in i's choice set.

In the case of the top-n policy this reasoning is a little bit more involved. Recall that for any node k that is querying the network, each agent has the same probability of being queried by this node. Hence we find that the random variables that are 1 if $k \in C_i$ and 0 otherwise are iid random variables for different agents i. Consequently, we find that the values $S_i^M(G_i, k)$ for different $i \in V$ are iid as well. And therefore for one of the sybil nodes s (without all other sybil nodes making queries) it holds $S_i^M(G_i, s) \geq S_i^M(G_i, k)$ has the same probability for all nodes $k \in C_i$. However, we find that for two sybil nodes s_1 and s_2 it holds $S_i^M(G_i, s_1)$ and $S_i^M(G_i, s_2)$ are not independent. In fact, it holds

$$\mathbb{P}\left(s_2 \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\} \mid s_1 \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right) \leq \mathbb{P}\left(s_2 \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right).$$

By Bayes theorem and the iid assumption we now know that it must hold

$$\begin{aligned}\mathbb{P}\left(\{s_1, s_2\} \subset \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right) \leq & \mathbb{P}\left(s_1 \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right) \cdot \mathbb{P}\left(s_2 \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right) \\ = & \mathbb{P}\left(s \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right)^2.\end{aligned}$$

Because for two different honest nodes $i, l \in V \backslash \{j\}$ $S_i^M(G_i, s_1)$ and $S_l^M(G_l, s_2)$ are independent, it also holds

$$\begin{aligned}&\mathbb{P}\left(s_1 \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}, s_2 \in \operatorname*{argmax}_{C_l' \subset C_l, |C_l|=n} \left\{S_l^M(G_l,k) \mid k \in C_l\right\}\right) \\ =& \mathbb{P}\left(s_1 \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right) \cdot \mathbb{P}\left(s_2 \in \operatorname*{argmax}_{C_l' \subset C_l, |C_l|=n} \left\{S_l^M(G_l,k) \mid k \in C_l\right\}\right).\end{aligned}$$

Lastly, it holds by the iid assumption of $S_i^M()$ that for any $s \in S$

$$\mathbb{P}\left(s \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right) = \mathbb{P}\left(s \in \operatorname*{argmax}_{C_l' \subset C_l, |C_l|=n} \left\{S_l^M(G_l,k) \mid k \in C_l\right\}\right)$$

is the same for all $i, l \in V$ and therefore

$$\mathbb{P}\left(s \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right) \cdot \mathbb{P}\left(s \in \operatorname*{argmax}_{C_l' \subset C_l, |C_l|=n} \left\{S_l^M(G_l,k) \mid k \in C_l\right\}\right) = \mathbb{P}\left(s \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right)^2$$

$$\mathbb{P}\left(\{s_1, s_2\} \subset \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right) \leq \mathbb{P}\left(s_1 \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}, s_2 \in \operatorname*{argmax}_{C_l' \subset C_l, |C_l|=n} \left\{S_l^M(G_l,k) \mid k \in C_l\right\}\right).$$

from which it follows that it is more strategic for any sybil attacker to attack the network in such a way that every of its sybil identities will attack a different honest node in the network. Both in the case of the winner-takes-all allocation policy as well as the top-n policy.

Corollary 4.1.1.
The largest expected profit attacker j can make from a sybil attack is smaller in the case of the winner-takes-all policy than in the case of the top-n policy. They're given by

$$\sum_{s \in S \cup \{j\}} \tilde{\omega} \cdot \mathbb{P}\left(s \in \operatorname*{argmax}_{k \in C_i}\left(S_i^M(G_i,k)\right)\right) = (|S|+1) \cdot \tilde{\omega} \cdot \mathbb{P}\left(s \in \operatorname*{argmax}_{k \in C_i}\left(S_k^M(G_i,C_i))\right)\right),$$

$$\sum_{s \in S \cup \{j\}} \tilde{\omega} \cdot \mathbb{P}\left(s \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_i^M(G_i,k) \mid k \in C_i\right\}\right) = (|S|+1) \cdot \tilde{\omega} \cdot \mathbb{P}\left(s \in \operatorname*{argmax}_{C_i' \subset C_i, |C_i|=n} \left\{S_k^M(G_i,C_i) \mid k \in C_i\right\}\right).$$

We should clarify the difference between proposition 4.1.1 and corollary 4.1.1. In proposition 4.1.1 we state that regardless of how the attacker distributes their sybils' queries over the network, i.e. regardless of the partition S' of the network, the expected amount work the attacker can consume is larger for the top n policy than for the winner-takes-all policy. In remark 4.1.1 we stated that the attacker can maximise the expected amount of work they can consume by distributing their sybils' queries such that no two sybils query the same node. Corollary 4.1.1 then concluded that this maximal amount of work is also larger for the top n policy than for the winner-takes-all policy by the same logic as in proposition 4.1.1.

Next, we show that the inverse inequality holds for the rank-weighted distribution policy and the winner-takes-all policy.

Remark 4.1.2.
As above, in the case of the rank-weighted distribution policy, the largest possible expected profit a sybil attacker can make in a single round is maximised by all sybil nodes attacking different honest nodes, i.e. for any partition $\{S_i \subset S \mid i \in V_i \setminus \{j\}\}$ it should hold $|S_i| \leq 1$ f.a. i.

Proposition 4.1.2.
The largest possible expected profit a sybil attacker can make is larger in the case of the winner-takes-all policy than in the case of the rank-weighted distribution policy.

Proof. In the case of the rank-weighted distribution policy, the largest possible expected profit a sybil attacker can make in a single round is given by $(|S|+1)\cdot\tilde{\omega}$, which can only be obtained if the choice sets of all honest nodes attacked by sybil nodes are empty, i.e. if no other honest nodes in the network query the attacked nodes. In formula this is given by

$$\mathbb{P}\left(|C_i| = 0 \text{ f.a.} |S_i| \neq 0\right).$$

We find that in that case the winner-takes-all policy returns the same profit for any accounting mechanism, as the only node in a choice set is automatically also the node with the highest accounting values and we conclude

$$\mathbb{P}\left(|C_i| = 0 \text{ f.a.} |S_i| \neq 0\right) \leq \mathbb{P}\left(\bigcap_{i \in V\setminus\{j\}} \operatorname*{argmax}_{k \in C_i}\left(S_i^M(G_i,k)\right) \in S_i\right).$$

□

Remark 4.1.3.
For any arbitrary, but fixed honest node i in an infinite network the probability of C_i being empty is given by

$$\mathbb{P}\left(C_i = \emptyset\right) = e^{-q}$$

and due to the iid assumption we also have for any partition S' that maximises the largest possible expected profit

$$\mathbb{P}\left(C_i = \emptyset \text{ f.a.} |S_i| \neq 0\right) = e^{-q+|S|+1}.$$

This is obviously decreasing as $|S|$ increases.

To recap we have that the probabilities of a sybil attack returning a profit of $(|S|+1)\cdot\tilde{\omega}$ for the 3 different allocation policies in question satisfy

$$\begin{aligned} e^{q+|S|+1} &= \mathbb{P}\left(\text{Rank-weighted distribution policy returns profit of } (|S|+1)\cdot\tilde{\omega}\right) \\ &\leq \mathbb{P}\left(\text{Winner-takes all policy returns profit of } (|S|+1)\cdot\tilde{\omega}\right) \\ &\leq \mathbb{P}\left(\text{Top-n policy returns profit of } (|S|+1)\cdot\tilde{\omega}\right). \end{aligned}$$

At this stage it may seem as though the rank-weighted distribution policy is most sybil resistant. Or at least has a stricter upper bound on the probability of a sybil attack returning its maximum profit. However, recall that we were investigating the effect of sybil attacks when scaled ($n \to \infty$). We will now take this into account when comparing the winner-takes-all policy and the rank-weighted distribution policy.

Proposition 4.1.3.
If for a sybil attack σ_j^n we let $n \to \infty$ then the maximum expected profit for the rank-weighted distribution policy converges to

$$(|V|-1)\cdot\tilde{\omega},$$

while for the winner-takes-all policy we obtain

$$(|V|-1)\cdot\tilde{\omega}\cdot\mathbb{P}\left(s=\operatorname*{arg\,max}_{k\in C_i}\left(S_i^M(G_i,k)\right)\right)$$

Proof. In the case of the rank-weighted distribution policy, a sybil attacker can scale their sybil region to infinity and attack every node in the honest region of the network with arbitrarily many of its sybil identities. At this point the expected amount of work the attacker can obtain from an attacked node converges to $\tilde{\omega}$ as it holds for $n, m \in \mathbb{N}$ variable with $n-m$ constant.

$$\frac{\sum\limits_{i=1}^{m} i}{\sum\limits_{i=1}^{n} i} \xrightarrow{n>m\to\infty} 1$$

As an example, take an honest node i with $C_i = \{k\}$ that is being attacked by a subset $S' \subset S$ with $|S'| \to \infty$. Let's assume that $S_i^M(G_i,k) > S_i^M(G_i,s)$ f.a. $s \in S'$. Then the sybil attacker obtains in a single round for different sizes of S' the results seen in table 4.1 below.

$\lvert S'\rvert = 1$:	$\lvert S'\rvert = 2$:	$\lvert S'\rvert = 3$:	$\lvert S'\rvert = 4$:
$\frac{\tilde{\omega}}{3}$	$\frac{3\cdot\tilde{\omega}}{6}$	$\frac{6\cdot\tilde{\omega}}{10}$	$\frac{10\cdot\tilde{\omega}}{15}$

Figure 4.1: Return of a sybil attack on a single node given different partition sizes.

Hence, we see that the profit converges to $\tilde{\omega}$ for $|S'| \to \infty$. This of course holds true for any other honest node, regardless of the number of nodes in its choice set, which yields a profit that converges to $\tilde{\omega}\cdot(|V|-1)$ if the sybil attackers attack all nodes i in V with $|S_i| \to \infty$.

In other words an attacker can simply "flood" the network with queries until it obtains $\tilde{\omega}$ from all participating honest nodes. This strategy is not feasible for the winner-takes-all policy as the attacker querying several nodes does not increase its chances of receiving work. □

Hence, we conclude that while the rank-weighted distribution policy may yield a smaller maximum expected profit for any finitely large sybil attack it is outperformed by the winner-takes-all allocation policy for $|S| \to \infty$. Because the winner-takes-all policy is a special case of the top n policy 4.1.3 holds for the top n as well and we obtain the following inequality

$$(|V|-1)\cdot\tilde{\omega}\cdot\mathbb{P}\left(s \in \operatorname*{arg\,max}_{C_i'\subset C_i,|C_i|=n}\left\{S_k^M(G_i,C_i)\mid k\in C_i\right\}\right) \leq (|V|-1)\cdot\tilde{\omega}.$$

Therefore we can conclude that the allocation policy that is most resistant to large scale sybil attacks is given by the winner-takes-all policy.

We have determined that the winner-takes-all allocation policy is the most resistant to large-scale sybil attacks, however it should be mentioned that the winner-takes-all also has a drawback relative to other policies. It strongly restricts data distribution to a single node per round and might lead to unfair or inefficient distribution of data in the network. The more agents an allocation policy serves the more effective the filesharing, but also the more susceptible it is to sybil attacks. Here, we find that its strength comes with a flaw.

At this point we should mention that the number of allocation policies that have been investigated in this section has been rather limited and seeing as there are infinite possible allocation policies these results are far from conclusive. There may be even far more effective allocation policies. However, we could not think of any during our research. We do think however, that we have analysed most reasonable allocation policies.

Note that we removed the banning policy from the list of viable options for allocation policies in our model, because it leads to an inefficient distribution of data among nodes in the network. The reason for this is that the banning policy will lead to some nodes not contributing to any of the nodes in their choice set, despite their willingness to seed. This will actually lead to less data being shared and somewhat defeats the purpose of filesharing networks, or at least is too strict.

Otte el al. (2016) introduce the strict winner-takes-all allocation policy in which the highest-ranking node is always served, but if all nodes in the choice set have the same accounting values then no node is served. The advantage to this allocation policy is that it is very resistant to sybil attacks as any node with accounting value 0 will never be served. Hence, the strict winner-takes-all policy will prevent the largest expected amount of work a sybil attacker can consume from converging to infinity. This solves the problem of sybil attacks, but has the same drawback as the banning policy, namely the fact that it very strictly limits the flow of data in the network.

At this point we have investigated our allocation policies in terms of their resistance to sybil attacks. However, we have not yet compared our allocation policies with respect to another metric, namely their resistance to lazy-freeriding. We would like to analyse, given a set of fixed accounting mechanisms, which of the allocation policies above are most successful in mitigating lazy freeriding and facilitating a fair distribution of data in a network. For this we simulate a network following the interaction model outlined in 4.1.1. In the experiments below, we simulate a network of only honest agents and determine which of the allocation policies above yield the most efficient data distribution and successfully mitigate lazy-freeriding.

4.1.3. Experimental Evaluation

We simulate a network with 100 honest nodes, query probability 0.7 and $\tilde{\omega} = 1$. We evaluate a number of different accounting mechanisms, such as the personalised PageRank algorithm [28], a degree-based accounting mechanism, discussed in example 3.3.1 and an accounting mechanism that assigns all nodes 0s. For allocation policies, we investigate the winner-takes-all policy, top-n (with and without distribution), whereby we choose $n = 4$, as well as the rank-weighted distribution policy. We run our simulation for 1000 rounds and determine the up- to download ratios of nodes in the network. The code to replicate any of these experiments are available on GitHub[1].

It might seem counter-intuitive to assume only honest nodes in such a simulation, as we are investigating which allocation policy will effectively prevent lazy-freeriding. However, we find that we do not need to actively simulate freeriders. Due to the Bernoulli distribution determining which nodes will query other nodes in the network for data and the uniformity of their choice over the whole network some agents will automatically not be queried for data as much as others. The number of queries a node will obtain as well as the number of queries a node will make will then follow an approximate Binomial distribution which means that

[1] `https://github.com/alexander-stannat/Msc-Thesis`

the ratio of queries made to queries received will have a fairly large standard deviation, leading to nodes that have received far fewer queries than they have made, and vice versa. We consider this an effective simulation of freeriders and altruists. We obtain the following distributions in figure 4.2 for the amount of queries made and received.

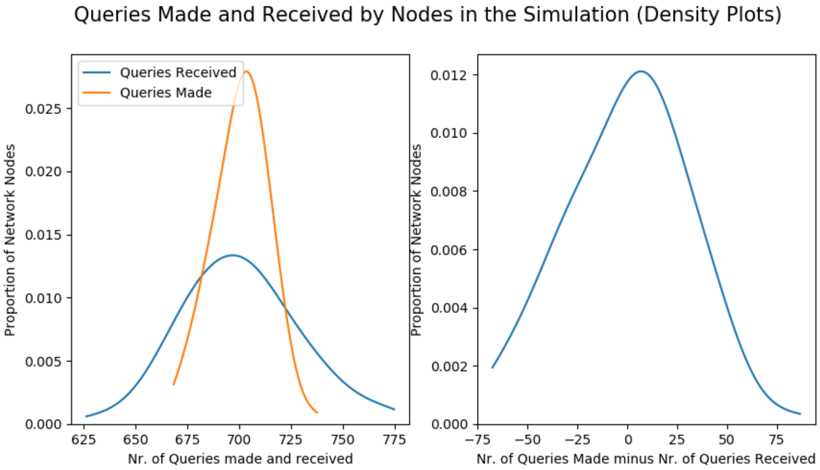
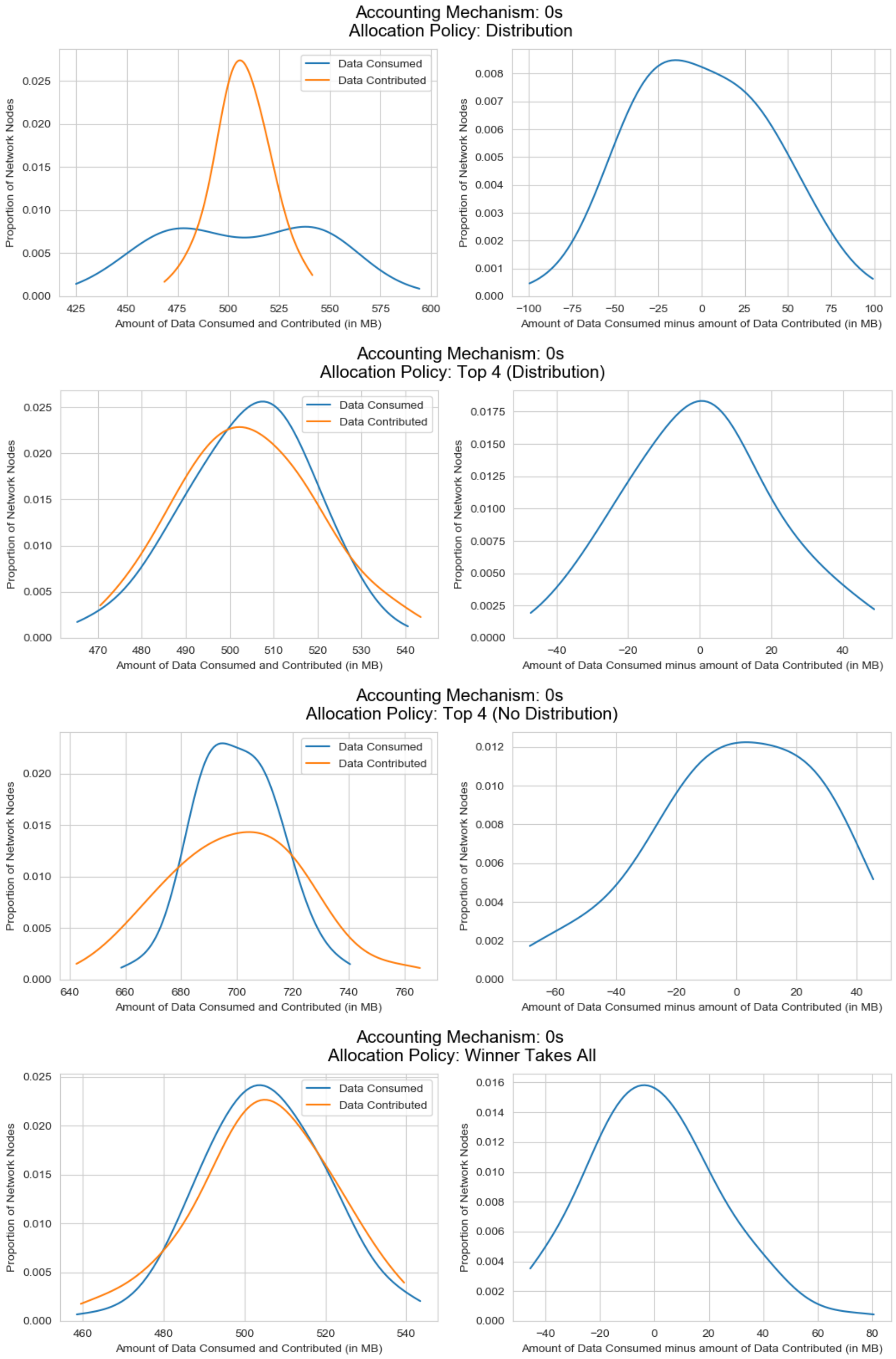


Figure 4.2: Results of simulated interaction model with 100 nodes, 0.7 query probability and 1000 rounds

In the graphs above we also find that the absolute values of queries nodes have received minus the queries they have made varies from values as small as -100 to values as large as 75. This indicates that the spectrum of altruism in the network is very wide with approximately as many nodes with a positive net value in number of queries as nodes with a negative net. Hence, we can conclude that we have simulated a fair number of (attempted) lazy freeriders and cooperators.

Next we investigate which different allocation policies are best in achieving a fair distribution of data in the network, i.e. which reward cooperative behaviour the most, while preventing lazy freeriding. For this we run the same network simulation for the accounting mechanisms and allocation policies mentioned above. The code to these simulations can be found on GitHub as well [2] We obtain the following set of data distributions.

[2] https://github.com/alexander-stannat/Msc-Thesis



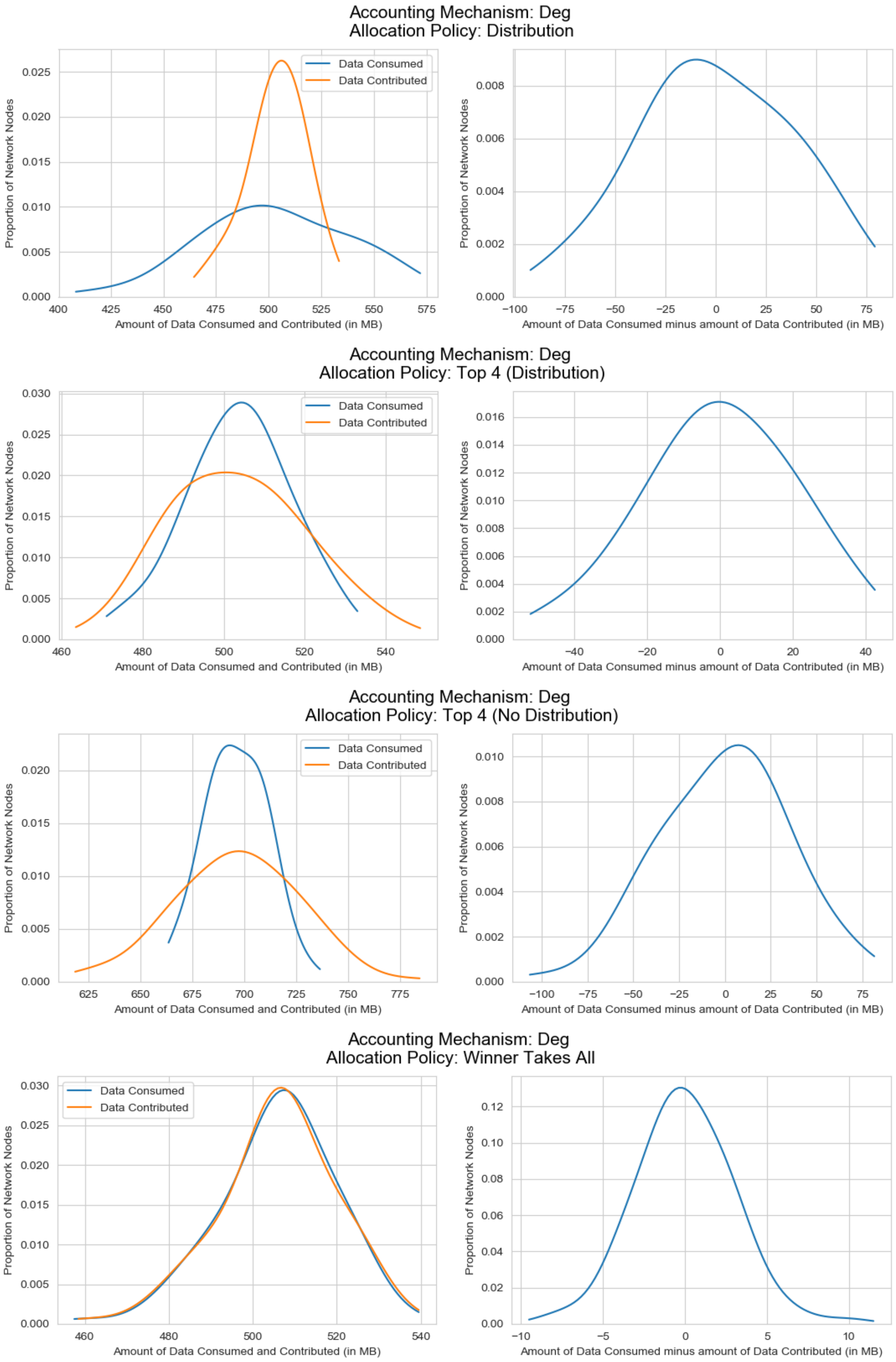
Accounting Mechanism: 0s
Allocation Policy: Distribution

Accounting Mechanism: 0s
Allocation Policy: Top 4 (Distribution)

Accounting Mechanism: 0s
Allocation Policy: Top 4 (No Distribution)

Accounting Mechanism: 0s
Allocation Policy: Winner Takes All

Figure 4.3: Simulation values of the Interaction Model given $S_i^M(G_i, j) = 0$.



Accounting Mechanism: Deg
Allocation Policy: Distribution

Accounting Mechanism: Deg
Allocation Policy: Top 4 (Distribution)

Accounting Mechanism: Deg
Allocation Policy: Top 4 (No Distribution)

Accounting Mechanism: Deg
Allocation Policy: Winner Takes All

Figure 4.4: Simulation values of the Interaction Model given $S_i^{Deg}(G_i, j)$.

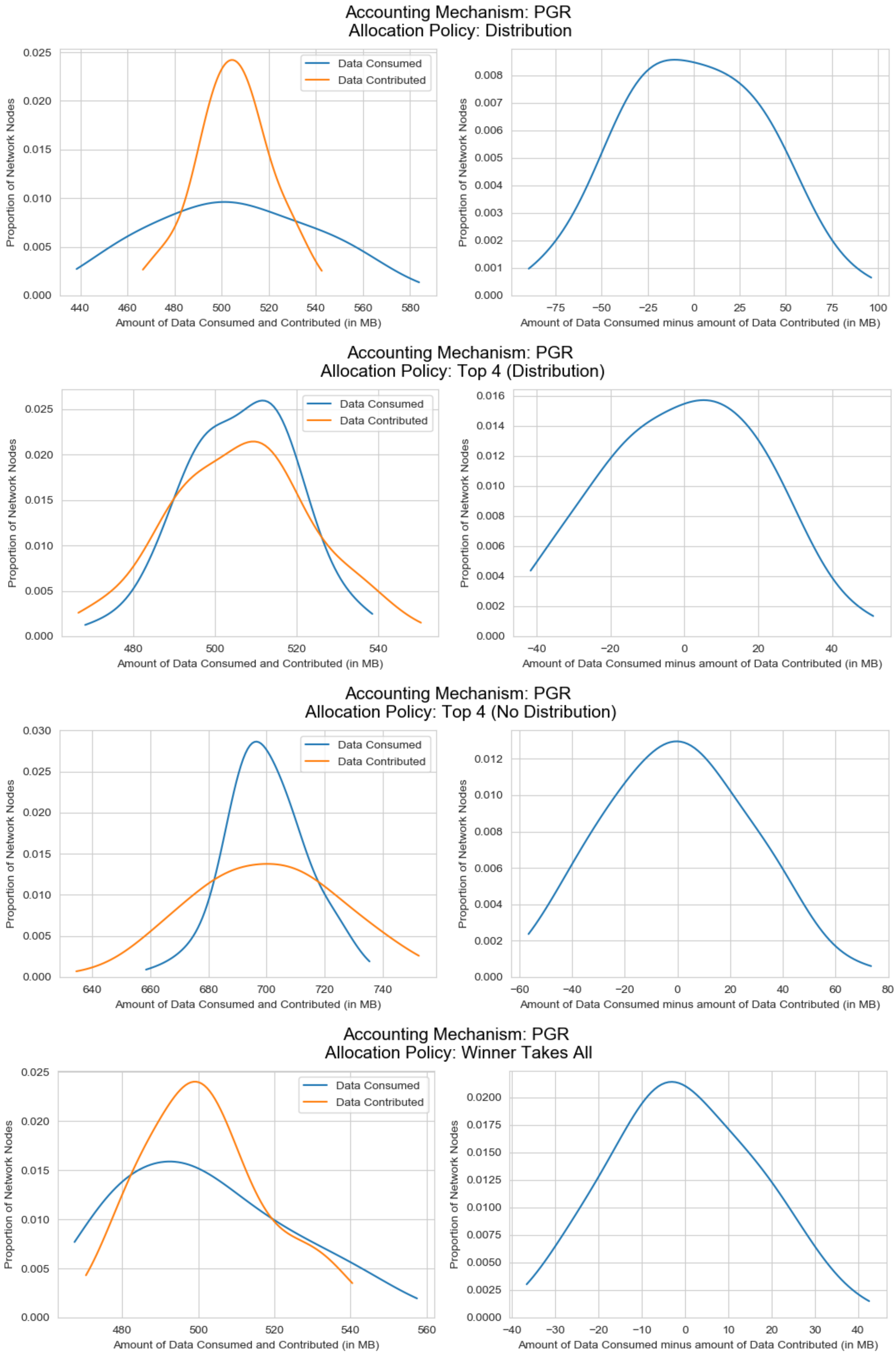


Figure 4.5: Simulation values of the Interaction Model given $S_i^{PGR}(G_i, j)$.

For the simplest accounting mechanism that assigns 0's to all nodes in the network, we find that the distribution policy leads to a rather wide-ranging distribution of net contributions made to the network, which implies that the distribution policy is not very effective in preventing or mitigating lazy freeriding. Additionally, we see that the distributions of contributions and consumption are quite different. This means the distribution policy leads to an somewhat inefficient distribution of data in the network and does not reward contributions effectively. The top-4 (distribution) policy does a much better job at preventing lazy freeriding than the distribution policy as we can see from the relatively small variance of the distribution of net consumption in the network. It also leads to a fairer distribution of data. The top-4 (no distribution) policy is even less effective at preventing lazy freeriding and facilitating a fair distribution of data. Lastly, we find that the winner-takes-all policy returns quite nice results for the distribution of net consumption with the exception of two outliers. The distributions of consumption and contribution are also not too different and we conclude that the best allocation policy for the all 0s accounting mechanism is in fact the winner-takes-all policy.

For the degree accounting mechanism we find that the distribution policy is very ineffective at punishing freeriders and does not distribute data fairly at all. The top 4 policy without distribution performs equally badly, as can be seen from a rather larger variance in net contributions and unequal distributions of contribution and consumption. The top 4 policy with data distribution is much stronger in both regards with a relatively small variance in the net contributions and similar graphs on the left as well. However, the winner-takes-all policy is by far the best with a extremely small variance in net contributions and almost equal distributions of consumption and contribution. Hence the winner-takes-all policy again is the most effective both in terms of preventing lazy freeriding as well as in facilitating cooperativeness.

In the case of the personalised PageRank accounting mechanism we find that the distribution policy returns a distribution similar to the distributions we saw for different accounting mechanisms. The fact that the distribution policy always returns very similar distributions of data in the network lies in the fact that every node in the choice set is always served and only the amount the it receives varies depending on its accounting value. The same holds for the top 4 (distribution policy). Although we do observe some differences between its distributions. These may, however be attributed to the randomness of the simulation. As before, the winner-takes-all policy outperforms all other policies, both in preventing freeriding and in rewarding contributions. We note that in terms of rewarding contributions it is outperformed by the top-4 (distribution) policy, albeit by not too much. We conclude this section by noting that the experimental results support our hypothesis and that the winner-takes-all policy is a good choice for the upper interaction model.

We realise that the set of allocation policies that we investigate is rather limited and that there may be much better allocation policies out there. A possible superior alternative to the winner-takes-all policy could be given by

$$A_i(S_i^M(G_i), C_i) := \left\{ j \in C_i \mid S_i^M(G_i, j) > 0 \right\},$$

where every node in $A_i(S_i^M(G_i), C_i)$ receives

$$\tilde{\omega} \cdot \frac{S_i^M(G_i, j)}{\sum\limits_{k \in C_i} S_i^M(G_i, k)}.$$

The reason this allocation policy may be superior to the winner-takes-all policy is that it prevents large-scale sybil attacks, due to the fact that it only serves nodes with accounting values greater than 0, and therefore sybil nodes will not be served as much. At the same time it does not restrict the distribution to a single node the way the winner-takes-all policy does. For time reasons we did not ffurther investigate this policy.

Recall that it was the goal of this chapter to define the cost and profit of a sybil attack and so far we have only covered the cost of a sybil attack in definition 4.1.2. The reason we took this "detour" to discuss an interaction model and allocation policies is so that we could define the profit of a sybil attack, which we will do now with the upper results on allocation policies in mind.

Definition 4.1.3 (Sybil Attack Profit).
For simplicity of formula we relable $X_{ij} := S_i^M(G_i, j)$ and define the random vectors $X_i := (X_{i1}, \dots, X_{in})$ for $i \in V$. We define the random variable Y_{ik} as the indicator function which is 1 if $X_{ik} \geq X_{ij}$ f.a. $j \in C_i$ and $k \in C_i$ and zero else. We then define $G'^{(1)}$ as the work graph after the attacker and their sybils have consumed the work they were elligible for after the attack. This continues round for round yielding work graphs $\left(G'^{(n)}\right)_{n \in \mathbb{N}}$. Then we define the expected amount of work node j can consume after sybil attack σ_j^n as

$$\omega_+^n = \sum_{n \in \mathbb{N}} \tilde{\omega} \cdot \mathbb{E}\left[\sum_{i \in V' \setminus \{j, s_{j1}, \dots, s_{j|S|}\}} \sum_{l=1}^{|S|} Y'^{(n)}_{is_{jl}} + Y'^{(n)}_{ij} \right].$$

At this point, our assumption of an infinite network from 4.1.1 comes into play. Without this assumption there would be a possibility that a node j with $S_i^M(G_i, j) = 0$ may be served by node i in a particular round, if it is the only node in i's choice set. Therefore, we find that for any node that queries another node with probability q in every round, the upper sum will be infinite. In order to curb this, assumed that while V_i is obviously always finite, there are infinite nodes $u \in V$ outside of i's subjective work graph, all with $S_i^M(G_i, u) = 0$ that will make queries to other nodes following the same paradigm as above. By this logic the choice set C_i will be of infinite size in every round and the probability of j being served in any given round is arbitrarily small. This results in the sum above being finite for any node in the network that does not "cheat".

Note that this specification has been completely neglected in the existing literature. In their research Seuken & Parkes have not specified the values ω_+^n and ω_-^n. However, it is an important one to make, as according to their definition it would always hold $\omega_+^n = \infty$. At least for most of the allocation policies discussed in section 3 and therefore any sybil attack would be strongly beneficial with respect to definition 4.1.1. They kept their definition of ω_+^n very vague, referring to it as "the amount of work that agent j or any of its sybils will be able to consume", but did not specify a time frame [23]. The same applies to ω_-^n.

Otte et al. (2016) solve this problem by assuming an allocation policy called the *strict winner-takes-all policy*, which serves the highest ranking node in the choice set, just like the winner-takes-all policy. However, if all nodes in the choice set have the same accounting values then the strict winner-takes-all policy doesn't serve anyone. We dismiss this allocation policy for the same reasons we disagree with the banning policy in 3.3.6, namely that it limits distribution of data in the network.

Now that we have definitions for both the cost and profit of a sybil attack we return to the definition of the benefit of a sybil attack, defined above in definition 4.1.1. Again, we say that a sybil attack is

Strongly Beneficial if $\omega_+^n > 0$ and $\omega_-^n = 0$ or if $\lim_{n\to\infty} \frac{\omega_+^n}{\omega_-^n} = \infty$,

Weakly Beneficial if $\omega_+^n > 0$ and $\omega_-^n > 0$ and $\exists c > 0 : \lim_{n\to\infty} \frac{\omega_+^n}{\omega_-^n} \geq c$.

We inverse these points to introduce sybil resistance.

Definition 4.1.4 (Sybil Resistance).
Let j be a malicious node perpetrating a sybil attack σ_j^n on the work graph G, resulting in work graph G'. Here n is variable. Now let $\omega_-^n > 0$ denote the cost of the sybil attack σ_j^n, according to definition 4.1.2 and let ω_+^n be the sybil attack profit defined in definition 4.1.3. Then we call a pair of accounting mechanism and allocation policy (S_i^M, A_i)

- **resistant against strongly beneficial sybil attacks** if

$$\forall j \in V_i \setminus \{i\} \, \forall (\sigma_j^n)_{n\in\mathbb{N}} \exists c > 0 : \lim_{n\to\infty} \frac{\omega_+^n}{\omega_-^n} \leq c,$$

- **resistant against weakly beneficial sybil attacks** if

$$\forall j \in V_i \setminus \{i\} \, \forall (\sigma_j^n)_{n\in\mathbb{N}} : \lim_{n\in\mathbb{N}} \frac{\omega_+^n}{\omega_-^n} = 0.$$

Note that sybil-proofness against weakly beneficial attacks is extremely restrictive and very hard to obtain by any form of accounting mechanism, while sybil-proofness against strongly beneficial attacks is easier to achieve and by our standards sufficient for the maintainance of a (mostly) cooperative, functioning network. The reason for this is that while an attacker can launch a sybil attack that returns a multiple of its investment, the idea is that in order for an attacker to leach infinite data, they also have to share infinite data. This means that any form of sybil attack will require some input into the network. No attacker can simply demand all resources in the network, leading to a complete shutdown. Instead, a weakly beneficial sybil attack still stimulates a network enough to maintain its existence.

Now that we have determined the cost and profit of sybil attacks, we should be able to determine for any pair (S_i^M, A_i) whether it is sybil-resistant or not. However, we remark that due to the probabilistic nature of the the sum in 4.1.3 it is impossible to compute for any generic setting making it impossible to determine how effective a sybil attack actually is. We need to renew the upper definitions in a way that makes them more easily computable.

This brings us to the next section where we reintroduce the profit and cost of sybil attacks in terms of accounting values, instead of work.

4.2. Redefining Cost & Profit in Terms of Accounting Values

Recall that accounting mechanisms were there to determine the standing of a node in the network and capture how much data an agent is elligible to consume. In a sybil attack it is the goal of the attacker to increase the accounting values of one or more nodes in the sybil region from the perspective of as many honest nodes as possible to be able to consume larger amounts of data from the network.

In order for us to be able to gauge how profitable an attack is, we introduce another pair of definitions of sybil attack rewards, which we denote $\omega_+^n(\text{rep})$ and $\omega_-^n(\text{rep})$. For uniformity we relabel the former definitions of cost and profit as $\omega_+^n(\text{work})$ and $\omega_-^n(\text{work})$.

In this setting, one might define the reward of an attack σ_j^n in terms of accounting mechanisms by the aggregate of accounting values all sybil nodes (including j) have collectively obtained through the attack.

$$\sum_{i \in V' \setminus \{j, s_{j1} \dots s_{jn}\}} \sum_{s \in \{j, s_{j1} \dots s_{jn}\}} S_i^M(G_i', s).$$

And we obtain the following definition of sybil attack profit.

Definition 4.2.1 (Sybil Attack Profit in Terms of Accounting Values)**.**
Given an objective and a subjective work graph $G := (V, E, w)$, $G_i := (V_i, E_i, w_i)$, let σ_j^n be a sybil attack of size $n \in \mathbb{N}$ with Sybil region $S = \{s_{j1}, \dots, s_{jn}\}$, whereby n is not fixed. Take $G' := (V', E', w')$ and $G_i' := (V_i', E_i', w_i')$ as defined above. We define $\omega_+^n(\text{rep})$ as the aggregate of accounting values that nodes in the sybil region collectively gain after it has carried out its attack. We obtain

$$\omega_+^n(\text{rep}) = \sum_{i \in V' \setminus \{j, s_1, \dots, s_n\}} \sum_{s \in \{j, s_1, \dots, s_n\}} S_i^M(G_i', s).$$

We now need to define $\omega_-^n(\text{rep})$. Just as in definition 4.2.1 this value needs to be equivalent to the earlier defined $\omega_-^n(\text{work})$, but in terms of accounting values. The reasoning for the definition of $\omega_-^n(\text{work})$ above was that we were trying to capture the amount of work invested into the network by a sybil attacker, i.e. the aggregate of the weights of the attack edges. In terms of accounting values, we can define this concept as the aggregated accounting values a sybil attacker has "earned" through their honest work, invested into the network.

Definition 4.2.2 (Sybil Attack Cost in Terms of Accounting Values)**.**
Given an objective and a subjective work graph $G := (V, E, w)$, $G_i := (V_i, E_i, w_i)$, let σ_j^n be a sybil attack of size $n \in \mathbb{N}$ with Sybil region $S = \{s_{j1}, \ldots, s_{jn}\}$, whereby n is not fixed. Take $G' := (V', E', w')$ and $G_i' := (V_i', E_i', w_i')$ as defined above.

Now we introduce a third graph $G'' = (V'', E'', w'')$ where $V'' = V$ and $w'' : V \times V \to \mathbb{R}$ with $w''(u, v) = w(u, v)$ f.a. $u, v \in V \setminus \{j\}$ and $w''(u, j) = \sum_{s \in S \cup \{j\}} w'(u, s)$ as well as $w''(j, u) = \sum_{s \in S \cup \{j\}} w'(s, j)$. Graphically, this means that we "collapse" all sybil nodes into a single node which we will label j again and all incoming and all outgoing edges from any sybil node into the honest region of the graph, are attached to j in G''.

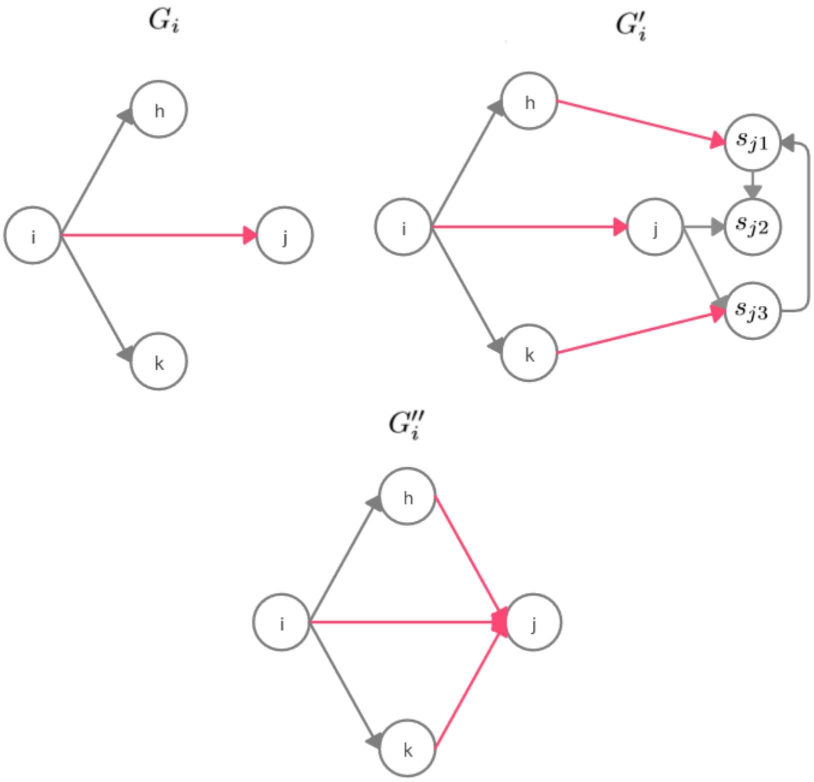


Figure 4.6: Example of Collapsing a Sybil Region

The point is that the aggregate of accounting values that the sybil attacker has gained should be compared to the accounting values that they're actually entitled to based on the actual work performed, i.e. the attack edges. All edges that do not enter or leave the sybil region are therefore disregarded and any increase in reputation that the sybil attackers may gain through the sybil-internal edges is dropped. In formula we then obtain the following value of sybil attack cost

$$\omega_-^n(rep) = \sum_{i \in V'' \setminus \{j\}} S_i^M(G_i'', j).$$

We should mention here that we keep the same definitions of sybil attack benefit and sybil resistance as in definitions 4.1.1 and 4.1.4 in both cases of sybil attack profit and cost. And we call sybil attacks strongly and weakly beneficial in terms of accounting values and in terms of work.

5 Representativeness of Accounting Mechanisms

Since we now have two different definitions for ω_+^n and ω_-^n, we must differentiate between strongly and weakly beneficial in terms of accounting values and in terms of work. Naturally, the question arises whether these two are equivalent. This is an important question as it is not feasible for us to determine the former ratio of cost to reward in units of work and therefore have to rely on the definition based on accounting values. However, it is the goal of any P2P filesharing network to ensure that there is no excessive data leakage, i.e. one wants to prevent sybil attacks that are strongly beneficial with respect to the work attackers can consume, while using accounting mechanisms as a proxy to enforce this. In order for accounting mechanisms to be able to do so, one requires an equivalence between the two.

Let us briefly recap the definitions determined in chapter 4. In summary, we obtained the following definitions of sybil attack costs and rewards.

$$\begin{aligned}
\omega_+^n(rep) &= \sum_{i\in V'\setminus\{j,s_1,\dots,s_n\}} \sum_{s\in\{j,s_1,\dots,s_n\}} S_i^M(G_i', s)\\
\omega_-^n(rep) &= \sum_{i\in V''\setminus\{j\}} S_i^M(G_i'', j)\\
\omega_+^n(work) &= \sum_{n\in\mathbb{N}} \tilde{\omega}\cdot\mathbb{E}\left[\sum_{i\in V'\setminus\{j,s_{j1},\dots,s_{jn}\}} \sum_{l=1}^{n} Y_{is_{jl}}'^{(n)} + Y_{ij}'^{(n)}\right]\\
\omega_-^n(work) &= \sum_{i\in V'\setminus\{j,s_{j1},\dots,s_{jn}\}} \sum_{s\in\{j,s_1,\dots,s_n\}} w'(s,i).
\end{aligned}$$

5.1. Incongruence of Sybil Attack Profits

However, we can come up with examples where the benefit defined in terms of work converges to ∞, but the ratio of accounting values does not. Inversely, there are examples of accounting mechanisms, for which there exists a strongly beneficial sybil attack in terms of accounting values, that is not strongly beneficial in terms of work.

Example 5.1.1.
Let $G = (V, E, w)$ be an arbitrary work graph and j a malicious node launching a sybil attack σ_j^n. Let every agent in the network have the same accounting mechanism S^M. Assume that there exists some $c > 0$ such that for any given subjective work graph G_i and any $i \in V$ it holds

$$\sum_{j\in V_i} S_i^M(G_i, j) = c.$$

Now, let $G' = G \downarrow \sigma_j^n$ be the work graph after the attack has been carried out, such that the accounting mechanism S^M satisfies

$$\lim_{n\to\infty} \sum_{k=1}^{n} S_i^M(G_i', s_{jk}) = c,$$

where $S = \{s_{j1}, \dots, s_{jn}\}$ are the sybil nodes created by j. We now find that it obviously holds

$$\lim_{n\to\infty} \sum_{k=1}^{n} S_i^M(G_i', s_{jk}) < \infty,$$

while we assume that in this attack there are finite attack edges and hence it follows

$$\lim_{n\to\infty} \frac{\omega(rep)_+^n}{\omega(rep)_-^n} < \infty.$$

However, as $\lim_{n\to\infty} \sum_{k=1}^{n} S_i^M(G_i', s_{jk}) = c$ it must follow that for every $\varepsilon > 0$ there exists $N \in \mathbb{N}$ such that for all $n \geq N$ it holds

$$\sum_{v \in V \setminus \{j, s_{j1}, \dots, s_{jn}\}} S_i^M(G_i', v) < \varepsilon.$$

And therefore it must follow that for any $\varepsilon > 0$ there exists an $N \in \mathbb{N}$ such that for all $n \geq N$ it holds $\mathbb{P}(Y_{is_{jk}} = 1) > 1 - \varepsilon$ f.a. $k = 1, \dots, n$. Hence, whenever either j or any of its sybil nodes query the honest node i for some work, they are very likely to be the highest ranking node in C_i and will therefore be served as much data as they want. It therefore holds

$$\lim_{n\to\infty} \omega_+^n(work) \geq \tilde{\omega} \cdot \lim_{n\to\infty} \mathbb{E}\left[\sum_{s \in \{j, s_{j1} \dots, s_{jn}\}} Y_{is}^{(1)} \right] = \infty.$$

Due to the finite attack edges we know that it must hold $0 < \omega_-^n(work) < \infty$ and it follows

$$\lim_{n\to\infty} \frac{\omega_+^n(work)}{\omega_-^n(work)} = \infty.$$

Such an accounting mechanism could be given by the PageRank algorithm, where a node j attacks the network with one edge connecting it to node i with a large edge weight and then creates many sybils which benefit from this attack edge. As the number of sybils grows, nodes in the sybil region will obtain an increasingly large proportion of the values. This is known as link spamming.

We can make this example more specific, by the following graph.

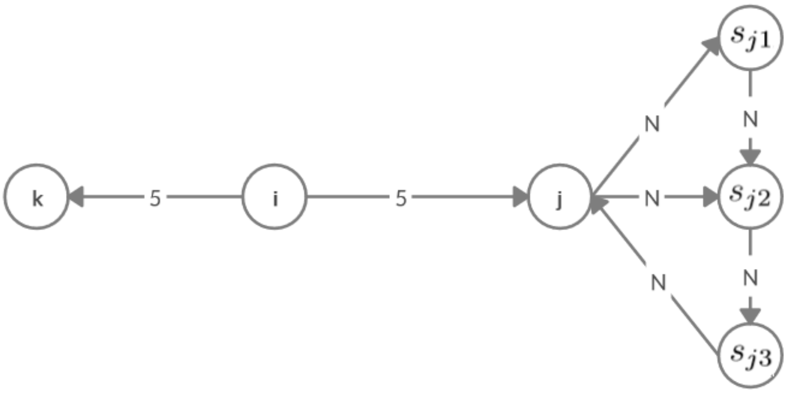


Figure 5.1: Example of Strongly Beneficial Sybil Attack in Terms of Work, but not Reputation

In this graph, i has the personalised pagerank algorithm as accounting mechanism S_i^M with a very low reset probability (≤ 0.0001) *[22]. j launches a sybil attack with 3 nodes and k is an honest node, having performed 5 units of work for i. j performs the same amount of work for i as an attack edge and then creates fake edges, connecting the sybils. These edges must have extremely high weights, in order for the sybil attack to be as effective as possible. Now, if i computes the personalised pagerank scores for all nodes in its subjective work graph we find that j and its sybil nodes have much higher reputation values than k. In this particular example for* $N = 1000$ *it would be*

$$k : 10^{-5}, j : 0{,}29, s_{j1} : 0{,}14, s_{j2} : 0{,}29, s_{j3} : 0{,}29.$$

Hence, it obviously holds $\frac{\omega_+^3(rep)}{\omega_-^3(rep)} = 4 < \infty$

Now, we want to compute $\frac{\omega_+^3(work)}{\omega_-^3(work)}$ *and we will show that this is infinite. Let s_{j1} now query some data from node i. It is guaranteed that s_{j1} is the highest ranking node in i's subjective work graph and will therefore be served some amount of data* $X(=5)$*, which changes the work graph into the one below.*

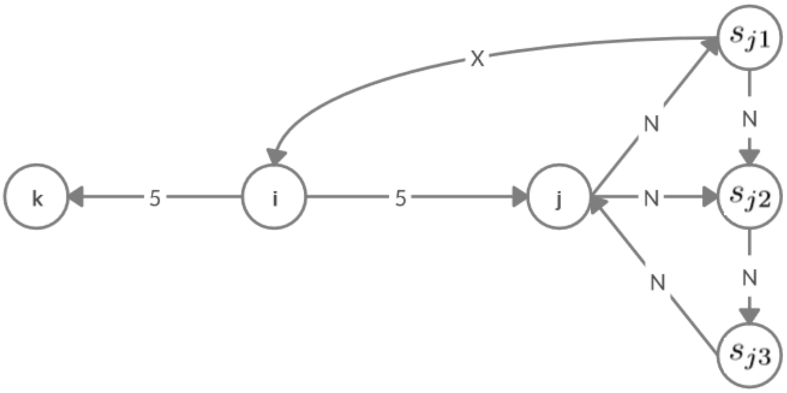


Figure 5.2: Example of Strongly Beneficial Sybil Attack in Terms of Work, but not Reputation

It now obviously holds $\tilde{\omega} \cdot \mathbb{E}[Y_{is_{j1}}] = 5$*. However, seeing as the weights of the fake edges are meant to be extremely high, s_{j1}'s leech of 5 units of work barely affects the pagerank values at all. We obtain the new accounting values*

$$k : 0{,}0001, j : 0{,}286, s_{j1} : 0{,}142, s_{j2} : 0{,}0286, s_{j3} : 0{,}286$$

Hence, it follows $\mathbb{P}(Y_{is_{j1}}^{(2)} = 1) = 1$ *and this continues for an arbitrarily long sequence of rounds. If at some point point the weight $w_i(s_{j1}, i)$ becomes so large that it begins to affect the accounting values of the other sybils, then j simply increases the weight of the fake edges again, and this game continues forever. Hence, we find that*

$$\frac{\omega_+^3(work)}{\omega_-^3(work)} = \infty.$$

We've shown that there exist accounting mechanisms such that there are sybil attacks that are strongly beneficial in terms of work, but not in terms of reputation.

Next, we introduce an example in which an attacker can accummulate infinite accounting values without necessarily gaining infinite work from it.

Example 5.1.2.
Let $S_i^M(G_i, k)$ be the accounting mechanism given by the number of shortest paths from i to every other node in the network that traverse node k, i.e. if $(SP_v^i)_{v \in V_i \setminus \{i\}}$ are the shortest paths connecting i and nodes $v \in V_i$ then

$$S_i^M(G_i, j) = \sum_{v \in V_i \setminus \{i\}} 1_{\{j \in SP_v^i\}}.$$

Now let node j be a sybil attacker creating attack σ_j^n such that $G' = G \downarrow \sigma_j^n$. We assume σ_j^n comprises one attack edge connecting j to another agent i and a large number of sybil nodes that all do some "work" for j.

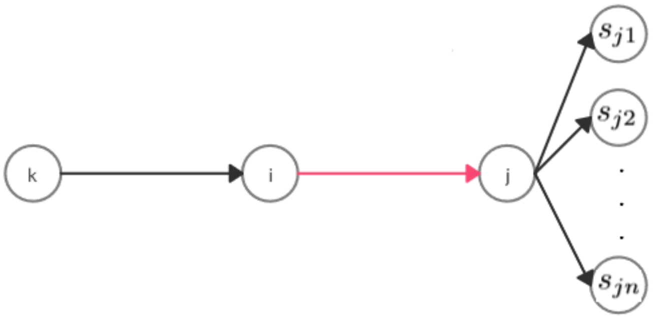


Figure 5.3: Example of Strongly Beneficial Sybil Attack in Terms of Reputation, but not Work

Consequently, node j will obtain the following accounting value from the perspective of i

$$S_i^M(G_i', j) = \sum_{v \in V_i' \setminus \{i\}} 1_{\{j \in SP_v^i\}} \geq \sum_{k=1}^{n} 1_{\left\{j \in SP_{s_{jk}}^i\right\}} = n.$$

Obviously, it then holds

$$\lim_{n\to\infty} \omega_+^n(rep) = \lim_{n\to\infty} \sum_{s \in \{j, s_{j1}, \dots, s_{jn}\}} S_i^M(G_i', s) = \infty,$$

while it also holds $\omega_-^n(rep) = 1$ and therefore it is $\lim_{n\to\infty} \frac{\omega_+^n(rep)}{\omega_-^n(rep)} = \infty$.

However, this does not imply that node j will have the highest (or one of the highest) accounting values in the network. As a matter of fact from k's perspective node i itself will have a higher accounting value than node j. This is because any path that traverses j must also traverse i, as j only has one attack edge. This may hold for many other nodes in the network as well and therefore the likelihood of node j being the highest ranking node in any other agent's choice set is not significantly close to 1. In the example given in figure 5.3 we find that if i and j both query node k for some work, i will be served and not j. In a large enough graph this will hold true for many other nodes. Hence it may follow $\lim_{n\to\infty} \frac{\omega_+^n(work)}{\omega_-^n(work)} < \infty$.

This is a rather important result to obtain. The point is that we can determine ω_+^n(rep) and ω_-^n(rep) and can therefore quite easily determine whether a sybil attack is strongly beneficial in terms of accounting values. However, the goal of an accounting mechanism is to represent the level of cooperativeness of a node in the network and prevent strongly beneficial sybil attacks in terms of work. The accounting values are therefore nothing but a proxy for a node's standing in the network. And when investigating sybil attacks we therefore only want to prevent $\lim_{n\to\infty} \frac{\omega_+^n(\text{work})}{\omega_-^n(\text{work})} = \infty$, while at the same time we are only able to determine $\lim_{n\to\infty} \frac{\omega_+^n(\text{rep})}{\omega_-^n(\text{rep})}$.

5.2. Defining Representativeness

This is rather problematic and we require the upper two concepts to be equivalent in order to evaluate the sybil-proofness of an accounting mechanism . If equivalence is not possible to achieve, we would like to find some restrictions on accounting mechanisms that make the latter stronger than the former. By this we mean that if a sybil attack is strongly beneficial in terms of reputation it must also be strongly beneficial in terms of work. Under this restriction we find that accounting mechanisms that are resistant to strongly beneficial sybil attacks in terms of reputation must also be resistant to them in terms of work. We call this property of accounting mechanisms *representativeness.*

Definition 5.2.1 (Representative)**.**
We say an accounting mechanism S^M is *weakly representative* if it holds for any work graph $G = (V, E, w)$ and

any sybil attack σ_j^n

$$\lim_{n\to\infty} \frac{\omega_+^n(rep)}{\omega_-^n(rep)} < \infty \implies \lim_{n\to\infty} \frac{\omega_+^n(work)}{\omega_-^n(work)} < \infty.$$

Subsequently, we call an accounting mechanism S^M *strongly representative* if it holds

$$\lim_{n\to\infty} \frac{\omega_+^n(rep)}{\omega_-^n(rep)} < \infty \iff \lim_{n\to\infty} \frac{\omega_+^n(work)}{\omega_-^n(work)} < \infty.$$

The question now arises, what requirements S^M must satisfy, in order for it to be weakly and/or strongly representative. We claim that in order for the upper definition of weak representativeness to be true there must exist some function f_{S^M} such that

$$f_{S^M}\left(\frac{\omega_+^n(\text{rep})}{\omega_-^n(\text{rep})}\right) = \frac{\omega_+^n(\text{work})}{\omega_-^n(\text{work})},$$

where $f_{S^M} : \mathbb{R}_{\geq 0} \to \mathbb{R}_{\geq 0}$ should be nondecreasing and well-defined, i.e. $f_{S^M}(x) < \infty$ f.a. $x < \infty$.

Additionally, for strong representativeness to be satisfied, f_{S^M} needs to satisfy $\lim\limits_{x\to\infty} f_{S^M}(x) = \infty$.

If for a given accounting mechanism S^M we know that such a function exists, then we can guarantee that a sybil-resistance in terms of accounting values implies sybil-resistance in terms of work. Conversely, we know that if f_{S^M} also satisfies $\lim\limits_{x\to\infty} f_{S^M}(x) = \infty$ then sybil-resistance in terms of work implies sybil-resistance in terms of accounting values.

Remark 5.2.1.
Note that for any arbitrary accounting mechanism that satisfies path-responsiveness it must already hold

$$\omega_-^n(work) = 0 \implies \omega_-^n(rep) = 0,$$

This is because any sybil attack with $\omega_-^n(work) = 0$ cannot have any attack edges, i.e. all edges added to the work graph by the attacker will be within the sybil region S and not connected to any outside nodes. This means that collapsing all sybil nodes will return an isolated node and by path-responsiveness an accounting value of 0. From this we can conclude that

$$\frac{1}{\omega_-^n(work)} = \infty \implies \frac{1}{\omega_-^n(rep)} = \infty$$

and obviously

$$\frac{1}{\omega_-^n(rep)} < \infty \implies \frac{1}{\omega_-^n(work)} < \infty.$$

Therefore we find that a function f_{S^M} mapping

$$f_{S^M}(\omega_+^n(rep)) = \omega_+^n(work)$$

with the same properties as mentioned above, suffices to ensure weak and strong representativeness. This f_{S^M} then denotes the maximum amount of data a set of nodes with a given aggregate of reputation values could leech without making any contributions anymore after the initial $\omega_-^n(work)$. If for a given accounting mechanism S^M we find that f_{S^M} is not well-defined then we conclude the given S^M is an inappropriate choice of accounting mechanism as it is not representative.

It should be noted here that for the most part it is impossible for us to explicitly determine a function f_{S^M} for any accounting mechanism in an arbitrary work graph with an undefined sybil attack. While this means that it may not be practically applicable, it does hold significant theoretical weight with respect to identifying strongly beneficial sybil attacks and/or sybil-proof accounting mechanisms. This leads us to the main result of this chapter summarised in the theorem below.

Theorem 5.2.1.
If an accounting mechanisms S^M does not allow for a strongly beneficial sybil attack σ_j^n with respect to accounting values to exist, i.e.

$$\forall \left(\sigma_j^n\right)_{n\in\mathbb{N}} : \lim_{n\to\infty} \frac{\omega_+^n(rep)}{\omega_-^n(rep)} < \infty$$

and it's (at least) weakly representative, then we find that it does not allow for any strongly beneficial sybil attacks in terms of work, i.e.

$$\forall \left(\sigma_j^n\right)_{n\in\mathbb{N}} : \lim_{n\to\infty} \frac{\omega_+^n(work)}{\omega_-^n(work)} < \infty$$

*In other words, an accounting mechanism that is **resistant to strongly beneficial sybil attacks in terms of accounting values**, and is at least **weakly representative**, is resistant to strongly beneficial sybil attacks in terms of work as well.*

An ideal accounting mechanism will satisfy both of these properties and if it doesn't, we will consider it an inappropriate choice.

This is a problem that has been widely disregarded in the literature so far, such as in [28]. So far the effectiveness of sybil attacks has only been researched for generic definitions of ω_+^n and ω_-^n, [23]. However, at closer inspection, we find that the more rigorous definitions introduced in chapter 4 are indeed required.

In order for us to make the upper theorem more concrete we introduce some examples below. We have already shown above in example 5.1.1 that the PageRank algorithm is not weakly representative and therefore it is obviously not strongly representative either.

Example 5.2.1 (Representativeness of BarterCast)**.**
Let S_i^M be the BarterCast accounting mechanism [12] for some honest node i in an arbitrary work graph $G = (V, E, w)$. Let j be a malicious node launching some arbitrary sybil attack $(\sigma_j^n)_{n\in\mathbb{N}}$, such that it holds

$$\lim_{n\to\infty} \frac{\omega_+^n(\text{rep})}{\omega_-^n(\text{rep})} < \infty.$$

Now, we know that for any node in the sybil region that consumes some data (j included) its accounting values from the perspective of some agents will decrease by at least the amount that is leeched. Hence, we know that no agent in the sybil region can consume ∞ data, and because $\omega_+^n(\text{rep}) < \infty$ there can only be finite nodes in the sybil region that gain data. Therefore it automatically follows

$$\lim_{n\to\infty} \frac{\omega_+^n(\text{work})}{\omega_-^n(\text{work})} < \infty.$$

However, while BarterCast may be weakly representative, it is not sybil resistant in terms of accounting values, as has been shown by Otte et al (2016) [19]. Therefore we can conclude that it is not a suitable accounting mechanism for the prevention of sybil attacks.

Example 5.2.2 (Representativeness of NetFlow)**.**
Let S_i^M be the Netflow (limited contribution) accounting mechanism [19] for some honest node i in an arbitrary work graph $G = (V, E, w)$. Let j be a malicious node launching some arbitrary sybil attack $(\sigma_j^n)_{n\in\mathbb{N}}$, such that it holds

$$\lim_{n\to\infty} \frac{\omega_+^n(\text{rep})}{\omega_-^n(\text{rep})} < \infty.$$

Then we already know by the same reasoning as in example 5.2.1 that it must already hold

$$\lim_{n\to\infty} \frac{\omega_+^n(\text{work})}{\omega_-^n(\text{work})} < \infty.$$

This is because the netflow mechanism is based on a variation of the BarterCast algorithm with an additional restriction introduced. Therefore we know that Netflow is weakly representative. However, Netflow does not satisfy strong representativeness. Due to the addition of node capacities, netflow achieves sybil resistance in terms of work, for any any sybil attack, regardless of its profit in terms of accounting values. Otte el al. (2016) have shown that no sybil attack will return an infinite benefit in terms of work for the attacker. We can think of plenty of sybil attacks which return an infinite benefit in terms of accounting values though and by this logic we know that it does not hold

$$\lim_{n\to\infty} \frac{\omega_+^n(\text{work})}{\omega_-^n(\text{work})} < \infty \Longrightarrow \lim_{n\to\infty} \frac{\omega_+^n(\text{rep})}{\omega_-^n(\text{rep})} < \infty.$$

Recall theorem 5.2.1 in which we stated that for sybil resistance to hold, an accounting mechanism needs to be at least weakly representative and resistant to strongly beneficial sybil attacks in terms of accounting values. Now that we have covered the concept of representativeness we look into what requirements need to be satisfied by accounting mechanisms for them to be resistant to sybil attacks in terms of accounting values and what requirements they must satisfy for them **not** to be resistant.

6

On the Impossibility of Sybil-Proofness

Recall that our overarching goal was to investigate the resistance of accounting mechanisms against strongly beneficial sybil attacks, in terms of work, of course. But we could only investigate sybil-proofness in terms of accounting values. This in combination with representativeness gave us a necessary requirement for sybil resistance in terms of work. In this section we will exactly define the kind of resistance we are looking for in a ranking algorithm, i.e. accounting mechanism and analyse some of the requirements accounting mechanisms must satisfy in order for them **not** to be resistant to such attacks. We critically examine some results from existing literature and expand on them.

We should mention here that many of the results below are phrased in a way that includes the possibility of misreports. This might seem redundant as we have already eliminated the possibility of misreports in chapter 3, however we will keep it as we are analysing existing literature. This means, we will maintain the notation of $w_i = (w_i^j(j,k), w_i^k(j,k))$.

6.1. Analysis of Existing Impossibility Results

Seuken and Parkes (2011) introduce a rather strong impossibility result [23], which we will recap here. They begin by defining the concept of single-report responsiveness.

Definition 6.1.1 (Single-Report Responsiveness)**.**
Let $G_i = (V_i, E_i, w_i)$ be the subjective work graph of agent i with nodes j and k, such that $(i, j) \in E$ with $w(i, j) > 0$ and no path in G_i connecting i and k. A report $(w_i^j(j,k), w_i^k(j,k))$ with $w_i^j(j,k) > 0$ yields a new subjective work graph G_i'. We call an accounting mechanism S^M *single-report responsive* if it then holds $S_i^M(G_i, k) < S_i^M(G_i', k)$.

This definition implies that a single (positive) report by a known neighbour about an unknown node will immediately result in an increase in the reputation of the unknown node. Note that this definition only covers two hops, i.e. if node k were to be further away from i then an accounting mechanism may be single-report responsive, despite node k not gaining any reputation from a report.

Differently put, if in the subjective work graph G_i there are nodes j, k, l such that $w_i(i, j) > 0$, $w_i^j(j,k), w_i^k(j,k) > 0$ and if a new report about an interaction is received by i with $w_i^k(k,l), w_i^l(k,l) > 0$ then it needn't hold $S_i^M(G_i, l) < S_i^M(G_i', l)$, which is an important distinction to make.

We will explain later why the single-report responsiveness axiom is in fact, a problematic one, leading to an impossibility result introduced by Seuken & Parkes (2011) [23].

The next definition introduced is called Independence of disconnected nodes.

Definition 6.1.2 (Independence of Disconnected Nodes)**.**
Given a subjective work graph $G_i = (V_i, E_i, w_i)$ and node $k \in V_i$ such that $w_i^j(j,k) = w_i^j(k,j) = w_i^k(k,j) = w_i^k(j,k) = 0$ for all $j \in V_i$. Let G_i' now denote the subjective work graph of i, with $V_i' = V_i \setminus \{k\}$ and $w'^i(j,l) = w^i(j,l)$ for all $j,l \neq k$. An accounting mechanism S^M is said to satisfy *independence of disconnected nodes* if $S_i^M(G_i, j) = S_i^M(G_i', j)$ for all $j \in V_i'$.

This means that removing a node that is not connected to any nodes from the work graph, will not affect the accounting values of any other nodes in the network.

The third relevant property is called symmetry, also referred to as anonymity.

Definition 6.1.3 (Symmetry)**.**
Given a subjective work graph G_i an accounting mechanism S^M is said to be *symmetric*, if for any graph isomorphism f with $G_i' = f(G_i)$ and $f(i) = i$ it holds

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

This means that from each individual's perspective, other agents' scores are invariant under relabelling. This is also a rather trivial necessity. Renaming agents in the network returns the exact same scores. This means that values returned by accounting mechanisms only depend on the structure of the subjective work graph and nothing else.

Seuken & Parkes (2011) introduce an impossibility result in which they prove the following theorem [23].

Theorem 6.1.1.
Every accounting mechanism that satisfies independence of disconnected agents, symmetry, single-report responsiveness and is misreport-proof there exists a passive strongly beneficial sybil attack (in terms of work).

Recall that we had introduced two definitions of misreport-proofness in chapter 3. The upper theorem relies on the former, definition 3.4.3, i.e. misreport-proofness on the choice set. However, seeing as general misreport-proofness is a stronger result, we can extrapolate this to definition 3.4.4.

Proof. The proof to this theorem is based on the following idea. Let G_i be the subjective work graph of agent i containing nodes $j,k \in V_i$, where node j is malicious and connected to node i, i.e. $w_i(i,j) > 0$. Let k not be connected to any other nodes in i's subjective work graph G_i at all.

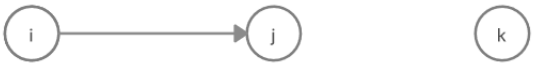


Figure 6.1: Step 1 in the Sybil Attack

Now, j may create a sybil identity s_{j1} which will not affect the scores any nodes in V_i, due to the independence of disconnected agents. If k now performs work for j then j is best off reporting this interaction to i, due to the misreport-proofness, and by single-report responsiveness it follows from this report that $S_i^M(G_i, k) < S_i^M(G_i', k)$. The symmetry condition now implies that one can apply a graph isomorphism f to G_i switching the labels of s_{j1} and k, i.e. $f(k) = s_{j1}$. From single-report responsiveness it now follows that $S_i^M(f(G_i), s_{j1}) > S_i^M(G_i, s_{j1})$ and due to misreport-proofness j does not suffer from the report on this edge.

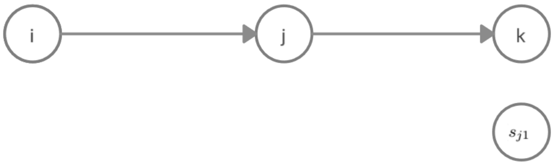


Figure 6.2: Step 2 in the Sybil Attack

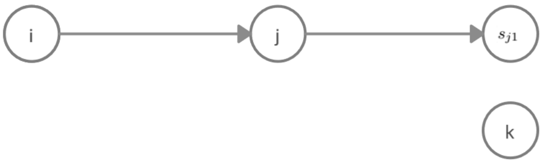


Figure 6.3: Step 3 in the Sybil Attack

The authors argue that because there was no actual work involved in this process, the attack given above is in fact a strongly beneficial sybil attack. They claim that therefore $\omega_-^n(\text{work}) = 0$ and that if $w(j, s_{j1})$ is large enough for s_{j1} to be chosen by i's allocation policy, it must hold $\omega_+^n > 0$. This implies $\frac{\omega_+^n(\text{work})}{\omega_-^n(\text{work})} = \infty$.

□

We disagree with the conclusion of this theorem. In fact, we believe that the attack mentioned above is not strongly beneficial, but weakly beneficial at best. The authors argue that no work is involved in the attack above as the only edge that was added to G_i by j is the edge (j, s_{j1}) which involved no work. We would disagree with this claim as the amount of work invested into a passive sybil attack should include all attack edges launched by j (see definition 4.1.3) and therefore be given by

$$\omega_-^n(\text{work}) = w(i, j),$$

which in the case of the proof above is greater than zero. Now, depending on the allocation policy of i, it will most likely hold $A_i(S_i^M(G_i), C_i) \in \{j, s_{j1}\}$, if j or s_{j1} query it, provided the edge weight $w^i(j, s_{j1})$ is large enough. However, this may only be true for finitely many rounds, after which i stops serving the attacker. Hence, we find that it will most likely hold:

$$\frac{\omega_+^n(\text{work})}{\omega_-^n(\text{work})} < \infty.$$

And therefore we have a weakly beneficial sybil attack at best.

Note that it also holds

$$\frac{\omega_+^n(\text{rep})}{\omega_-^n(\text{rep})} < \infty,$$

so long as $S_i^M(G_i', s_{j1}), S_i^M(G_i', j) < \infty$.

Hence, we argue that by our definition of sybil attack profit, theorem 1 from [23] is no longer valid. The problem with this theorem is that the authors were not rigorous enough in their definitions of ω_+^n and ω_-^n. This is the reason they were able to argue that $\omega_-^n = 0$. At closer inspection however, it should be clear that the edge (i, j) was created by the attacker and without it the attack would not be beneficial at all. We feel like it is clear that this edge should be weighted into ω_-^n. The authors argue as though the edge had already been part of the work graph and not created by the attacker for the sake of the attack. If that is the case then the authors should have specified that the attack above is only applicable in a very particular work graph, in which the upper edge already exists. Seeing as they did not do this, but instead argued on the grounds of an arbitrary work graph, we believe that this theorem above is not valid.

This has highlighted the necessity for our much more rigorous definitions of sybil attack cost and profit introduced in chapter 3, as the ambiguity of their definitions for ω_+^n and ω_-^n has lead to an inaccurate/flawed result. We have shown that the upper sybil attack was not actually strongly beneficial.

Another mistake made in this theorem is that absolutely no restrictions were made on the allocation policy of i. It was simply assumed that j could increase its sybils scores by enough for it to follow $A_i(S_i^M(G_i'), C_i') \in \{j, s_{j1}\}$. This may not necessarily be the case and so we would argue that the theorem above does not actually imply a beneficial sybil attack in terms of work, but only in terms of accounting values. And again, it really is just weakly beneficial with respect to accounting values as well, for the same reasons as given above. If the reader wants to verify this, we refer them to our definitions in chapter 5.

6.2. Improving Existing Impossibility Results

However, the existence of a strongly beneficial passive sybil attack under the conditions in theorem 6.1.1 is not yet disproven. Note that in the upper attack it may still be possible to add additional sybil nodes $s_{j2}, s_{j3}, \ldots$ such that $w_i(j, s_{jl}) > 0$. By single-report responsiveness all of these nodes obtain accounting values $S_i^M(G_i, s_{jl}) > 0$. By symmetry, it must hold $S_i^M(G_i, s_{jl}) = S_i^M(G_i, s_{jk})$ f.a. $l, k \in \mathbb{N}$ so long as $w_i(j, s_{jl}) = w_i(j, s_{jk})$ f.a. $l, k \in \mathbb{N}$.

6.2.1. Parallel-Report Responsiveness

If we now introduce one additional condition, we can prove that a strongly beneficial passive sybil attack does exist. For this we introduce a new definition called parallel-report responsiveness.

Definition 6.2.1 (Parallel-Report Responsiveness)**.**
Given a subjective work graph $G_i = (V_i, E_i, w_i)$ with choice set C_i and nodes j, k, l such that $w^i(i, j) > 0$ and no path connecting i and the two nodes k, l. If G_i' is the graph given by the single-report responsiveness to the edge report $w_i^j(j,k)' > 0$ and G_i'' is the subjective work graph given by G_i' combined with the onset of the edge $w_i^j(j,l)'' > 0$. We call an accounting mechanism S^M *parallel-report responsive* if it is single report responsive and the addition of a second report does not influence the value of S^M, i.e.:

$$S_i^M(G_i', k) \not> S_i^M(G_i'', k).$$

This means that if node j adds multiple sybil nodes, one after the other then the reputation of the sybils won't be influenced by the introduction of newer sybils. Sybils sharing the same node as perpetrator of a passive sybil attack do not have to "share" the increase in reputation gained.

This definition leads to a new theorem

Theorem 6.2.1.
Every accounting mechanism S^M that satisfies independence of disconnected nodes, symmetry, single-report responsiveness and parallel-report responsiveness as well as being misreport-proof has a passive strongly beneficial sybil attack in terms of accounting values.

Proof.
Let G_i be the subjective work graph of agent i containing agents $j, k \in V_i$, where j is the attacker (analogously to theorem 6.1.1). Now j launches a sybil attack σ_j^n creating n sybil identities $s_{j1}, \ldots, s_{jn}$ as indicated in figure 7.1. This yields a graph $G_i' = G \downarrow \sigma_j^n = (V_i', E_i', w_i')$ with $V_i' = V_i \cup \{s_{j1} \ldots s_{jn}\}$. Due to the independence of disconnected agents the scores of all agents in V_i have not changed. Since there is no edge connecting i to any nodes in $\{k, s_{j1}, \ldots, s_{jn}\}$ from i's perspective they are indistinguishable.

Now assume that k performs c units of work for j then due to misreport-proofness j is best off reporting the transaction honestly and by single-report responsiveness it will follow $S_i^M(G_i', k) > S_i^M(G_i, k)$. Due to the symmetry of S^M we can apply a number of graph isomorphisms $f_1, \ldots, f_n$ where f_l only switches the labels of nodes k and s_{jl}. Consequently, there exists a report that j can make about each of its sybil nodes such that $w_i^j(j, s_{jl}) = c$, leading to a graph $G_i^{(l)}$ where $S_i^M(G_i^{(l)}, s_{jl}) > 0$.

Now, due to *parallel-report responsiveness* we find that in a graph $\tilde{G}_i^{(l)}$ with $w_i^j(j, s_{j1}) =, \ldots, = w_i^j(j, s_{jl}) = c$ and $w_i^j(j, s_{jl+1}), \ldots, w_i^j(j, s_{jn}) = 0$ such that $S_i^M(\tilde{G}_i^{(l)}, s_{j1}) = \ldots = S_i^M(\tilde{G}_i^{(l)}, s_{jl}) > 0$ and $S_i^M(\tilde{G}_i^{(l)}, s_{jl+1}) = \ldots = S_i^M(\tilde{G}_i^{(l)}, s_{jn}) = 0$, adding a report $w_i^j(s_{jl+1}, j) = c$ leads to a graph $\tilde{G}_i^{(l+1)}$ with $S_i^M(\tilde{G}_i^{(l+1)}, s_{jl+1}) > S_i^M(\tilde{G}_i^{(l)}, s_{jl+1})$ and $S_i^M(\tilde{G}_i^{(l+1)}, s_{jr}) \geq S_i^M(\tilde{G}_i^{(l)}, s_{jr})$ f.a. $r \in \mathbb{N}_{\leq l}$. Now we find that by symmetry and the fact that all edges in the sybil region have the same weight, it must hold

$$\frac{\omega_+^n(\text{rep})}{\omega_+^{n+1}(\text{rep})} \leq \frac{n}{n+1}.$$

Because there is only one attack edge $w_i(i, j) > 0$, we find that $\omega_-^n(\text{rep})$ must be constant f.a. $n \in \mathbb{N}$. From this we conclude that in fact it holds

$$\lim_{n\to\infty} \frac{\omega_+^n(\text{rep})}{\omega_-^n(\text{rep})} = \infty.$$

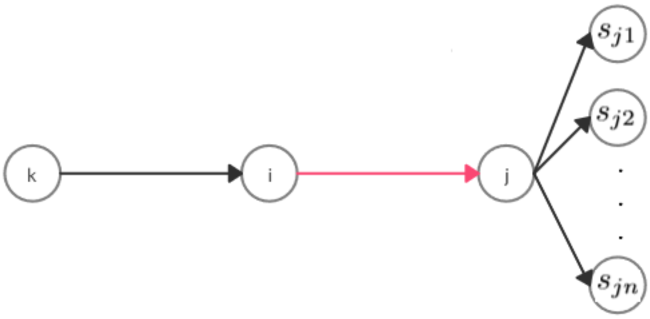


Figure 6.4: Sybil Attack on Parallel-report Responsive Accounting Mechanism

□

This theorem only returns a strongly beneficial sybil attack in terms of accounting values and not in terms of work performed. However, if the accounting mechanism additionally is weakly representative, it will imply a strongly beneficial sybil attack in terms of work as well.

Some confusion may arise because, while an additional sybil node may not reduce the reputation values of existing sybil nodes, due to parallel-report responsiveness, the sum of reputations may still converge to a finite value. For example one could imagine that it may hold $S_i^M(\tilde{G}_i^{(n)}, s_{jn}) = 2^{-n}$. Then it would follow $\sum_{n\in\mathbb{N}} S_i^M(\tilde{G}_i^{(n)}, s_{jn}) = 1$ and consequently $\lim_{n\to\infty} \frac{\omega_+^n(\text{rep})}{\omega_-^n(\text{rep})} < \infty$. This is prevented by the symmetry assumption. Because from i's perspective all sybils look the same their respective reputation values must be the same, which justifies our claim that $\frac{\omega_+^{n+1}(\text{rep})}{\omega_+^n(\text{rep})} \geq \frac{n+1}{n}$ Due to the symmetry assumption, it must hold that $S_i^M(\tilde{G}_i^{(n)}, s_{jl}) = S_i^M(\tilde{G}_i^{(n)}, s_{jk})$ f.a. $l, k \leq n$. Therefore, the return of the sybil attack in terms of accounting values is given by $n \cdot S_i^M(\tilde{G}_i^{(n)}, s_{jn})$ and it is therefore strongly beneficial if $S_i^M(\tilde{G}_i^{(1)}, s_{j1}) > 0$, which we know is true by single-report responsiveness.

Remark 6.2.1.
We have introduced the property of parallel-report responsiveness for good reason. Without it, the upper attack seen in figure 7.1 may not actually be strongly beneficial. To highlight this, we return to the example of the personalised PageRank accounting mechanism. S^{PGR} satisfies all properties required for theorem 6.2.1, except for parallel-report responsiveness.

Altman et al. (2005) have shown that the personalised PageRank algorithm satisfies symmetry [1]. The independence of disconnected nodes and single-report responsiveness properties are trivially satisfied as well. Lastly, we know that misreport-proofness is obviously satisfied, due to the TrustChain data structure underlying all accounting mechanisms discussed in here. However, the upper attack is not strongly beneficial in terms of accounting values as it must hold $\sum_{l=1}^{n} S_i^{\text{PGR}}(\tilde{G}_i^{(n)}, s_{jl}) = \sum_{l=1}^{n+1} S_i^{\text{PGR}}(\tilde{G}_i^{(n+1)}, s_{jl})$ f.a. $n \in \mathbb{N}$ and therefore $S_i^{\text{PGR}}(\tilde{G}_i^{(n+1)}, s_{jn+1}) \leq S_i^{\text{PGR}}(\tilde{G}_i^{(n)}, s_{jn})$ The lack of parallel-report responsiveness in PageRank leads to a non-strongly beneficial sybil attack in terms of accounting values.

From theorem 6.2.1, we can derive a slightly weaker definition of parallel-report responsiveness and derive an even stronger corollary.

Definition 6.2.2 (Weak Parallel-report responsiveness)**.**
Given a subjective work graph $G_i = (V_i, E_i, w_i)$ with nodes $j, s_{j1}, \ldots, s_{jn}$ such that $w^i(i, j) > 0$ and no path connecting i and the nodes $s_{j1}, \ldots, s_{jn}$. Let $G_i^{(1)}$ be the graph given by the single-report responsiveness to the edge report $w_i^j(j, s_{j1}) > 0$ and $G_i^{(2)}$ the subjective work graph given by $G_i^{(1)}$ combined with the onset of the edge $w_i^j(j, s_{j2}) > 0$. More generally, we define $G^{(l)}$ as the graph $G^{(l-1)}$ with the edge $w(j, s_{jl})^{(l)} > 0$. We call an accounting mechanism S^M *weakly parallel-report responsiveness* if it is single-report responsive and the additional reports yield an infinite sum, i.e.

$$\lim_{n \to \infty} \sum_{l=1}^{n} S_i^M(G_i^{(n)}, s_{jl}) = \infty.$$

If the accounting mechanism is also symmetric and all values $w(j, s_{jl})$ are equal then this implies $S_i^M(G_i^{(n)}, s_{jl})$ are all equal f.a. $l \leq n$. The upper definition then becomes equivalent to $S_i^M(G_i^{(n)}, s_{j1}) = \omega(\frac{1}{n})$.

Note that parallel-report responsiveness implies weak parallel-report responsiveness and is therefore a stricter condition. We obtain the following corollary.

Corollary 6.2.1.
Every accounting mechanism S^M that satisfies independence of disconnected nodes, symmetry, single-report responsiveness and weak parallel-report responsiveness, as well as being misreport-proof, has a strongly beneficial sybil attack in terms of accounting values.

Note that the upper theorem and corollary hold for active sybil attacks as well, because any passive sybil attack is also an active sybil attack, but not every active sybil attack is also a passive sybil attack. Active sybil attacks are generally stronger than passive sybil attacks as there can be more attack edges, but not every active sybil attack is more beneficial than a passive one. There may be cases in which an additional attack edge increases the cost/investment of a sybil attack without increasing the return proportionately.

In some cases it may be more beneficial to launch an active attack, distributing the attack edges over several nodes, while in others it may be advantageous to confine the attacking nodes to a single agent. This very much depends on the accounting mechanism at hand and the structure of the work graph. However, the main take-away is that if there exists a strongly beneficial passive sybil attack under certain conditions then there must also exist an active one. Consequently, if an accounting mechanism satisfies requirements that imply immunity to any strongly beneficial active sybil attack then there can also not be any passive sybil attacks. We can rewrite this as a lemma.

Lemma 6.2.1.
Let $G_i = (V_i, E_i, w_i)$ be an arbitrary subjective work graph with $j \in V_i$. Then for any passive sybil attack of arbitrary size n, σ_j^n there exists an active sybil attack $\tilde{\sigma}_j^n$ such that $\tilde{\omega}_+^n \geq \omega_+^n$. Additionally, it holds that every accounting mechanism S^M that is resistant to active strongly beneficial sybil attacks is also resistant to passive strongly beneficial sybil attacks in terms of accounting values.

This holds both for beneficial sybil attacks in terms of work and in terms of accounting values, which is why we simply wrote ω_+^n instead of ω_+^n (work) or ω_+^n (rep).

From this lemma we can derive a corollary to theorem 6.2.1 corollary and 6.2.1.

Corollary 6.2.2.
Every accounting mechanism S^M that satisfies independence of disconnected nodes, symmetry, single-report responsiveness and (weak) parallel-report responsiveness, as well as being misreport-proof, has an active strongly beneficial sybil attack in terms of accounting values.

We have now seen that any accounting mechanism satisfying the conditions of misreport-proofness, symmetry, independence of disconnected agents, as well as single-report responsiveness and parallel-report responsiveness is susceptible to strongly beneficial sybil attacks. Misreport-proofness is, in our case, always guaranteed, due to the reasons discussed in chapter 3. The condition of symmetry is also an appropriate condition and should be satisified by any sensible accounting mechanism. The condition "independence of disconnected agents" is also a very sensible property for any accounting mechanism to have and we will not contest it in here. However, we do have some reservations about the definitions of single-report responsiveness and parallel-report responsiveness, which weaken the importance of this theorem.

6.2.2. Extending the Model to Multiple Hops

While the upper two theorems may seem like very strong assertions, they are, in fact, rather weak. The definitions of single-report responsiveness and parallel-report responsiveness above are quite narrowly defined. The problem here is that they are only defined for agents in the work graph that are exactly two "hops" away from the node determining the accounting values. We would like to extend this definition to an arbitrary distance from the seed node. We now extend our definitions to multiple hops.

Definition 6.2.3 (Several-Hop Single-report Responsiveness)**.**
Given an arbitrary subjective work graph $G_i = (V_i, E_i, w_i)$, in which there exists a node $k \in V_i$ such that there is no path in G_i connecting i and k. Now, let G_i' be the same subjective work graph as G_i, but with a path of arbitrary length n given by $\{j_1, \dots, j_n\}$ added, such that there exists some $c > 0$ with $w_i(i, j_1) \geq c$, $w_i^{j_{l-1}}(j_{l-1}, j_l) \geq c$ f.a. $l \in \mathbb{N}_{\leq n}$ and $w_i^{j_n}(j_n, k) \geq c$. We say that an accounting mechanism S^M satisfies several-hop single-report responsiveness, if it holds

$$S_i^M(G_i', k) > S_i^M(G_i, k).$$

Definition 6.2.4 (Several-Hop Parallel-report Responsiveness)**.**
Given an arbitrary subjective work graph $G_i = (V_i, E_i, w_i)$, in which there exist nodes $j, k, l \in V_i$ such that there is no path in G_i connecting i and k, l and there exists a path of length n given by $\{j_1, \dots, j_n\}$, such that there exists some $c > 0$ with $w_i(i, j_1) \geq c$, $w_i^{j_{l-1}}(j_{l-1}, j_l) \geq c$ f.a. $l \in \mathbb{N}_{\leq n}$ and $w_i^{j_n}(j_n, j) \geq c$. Now, let G_i' be the same subjective work graph as G_i, but with i having received the report $w_i^j(j, k) > 0$ and let G_i'' be the subjective work graph G_i' with the additional report $w_i^j(j, l) > 0$. We say that an accounting mechanism S^M satisfies several-hop parallel-report responsiveness, if it holds

$$S_i^M(G_i', k) \ngtr S_i^M(G_i'', k).$$

Note that the definitions of (single-hop) single-report responsiveness and (single-hop) parallel-report responsiveness are special cases of several-hop single-report responsiveness and several-hop parallel-report responsiveness. Hence, the following corollary is a simple extrapolation of theorem 6.2.1.

Corollary 6.2.3.
Every accounting mechanism S^M that satisfies symmetry, independence of disconnected nodes, several-hop single-report responsiveness, several hop parallel-report responsiveness as well as being misreport-proof has a strongly beneficial (passive) sybil attack in terms of accounting values.

The proof to this is equivalent to the proof of theorem 6.2.1 with the only difference being that the sybil attack can be "further away". What we have now achieved is that the upper results are true for far more work graphs.

Analogously, we can extend the definition of weak parallel-report responsiveness to multiple hops as well and derive an equivalent corollary to corollary 6.2.1 for sybil attacks that are further away. Lastly, we can then also derive a corollary that is equivalent to corollary 6.2.2, which conclude this section.

6.2.3. Serial-Report Responsiveness

We have now proven the existence of a strongly beneficial sybil attack in terms of accounting values under the assumption of several-hop (weak) parallel-report responsiveness. Next, we introduce another requirement, based on which an accounting mechanism might not be resistant to strongly beneficial sybil attacks, which we call serial-report responsiveness.

Definition 6.2.5 (Serial-report responsiveness)**.**
Given a subjective work graph $G_i = (V_i, E_i, w_i)$ of agent i with nodes j, k, l such that there exists no path in G_i connecting i and k, l and there exists a path of arbitrary length n given by $\{j_1, \dots, j_n\}$ such that there exists some $c > 0$ with $w_i(i, j_1) \geq c$, $w_i^{j_{l-1}}(j_{l-1}, j_l) \geq c$ f.a. $l \in \mathbb{N}_{l\leq n}$ and $w_i^{j_n}(j_n, j) \geq c$. Now let G_i' be the same as G_i except for an added report $w_i^j(j, k) > 0$. Now, a several-hop single-report responsive accounting mechanism S^M will satisfy $S_i^M(G_i', k) > S_i^M(G_i, k)$. Let G_i'' be the same as G_i' with the additional report $w_i^k(k, l) \geq w_i^j(j, k) > 0$. We say that the accounting mechanism S^M is serial-report responsive if the following two conditions are satisfied

$$S_i^M(G_i'', l) \geq S_i^M(G_i', k)$$

and

$$S_i^M(G_i'', k) \geq S_i^M(G_i', k).$$

There is a reason we have defined parallel-report responsiveness and single-report responsiveness twice, but serial-report responsiveness only once. Single-report responsiveness and parallel-report responsiveness can be defined for single-hops and several-hops. Serial-report responsiveness, however cannot be defined for only a single hop. This is because the very definition itself implies several hops from the seed node i. A single-hop serial-report responsiveness definition does not make sense in this situation. This leads us to our next theorem.

Theorem 6.2.2.
Every accounting mechanism S^M that satisfies independence of disconnected nodes, symmetry, several-hop single-report responsiveness and serial-report responsiveness as well as being misreport-proof has a passive strongly beneficial sybil attack in terms of accounting values.

Proof.
Let G_i be the subjective work graph of agent i containing agents $j, k \in V_i$ and a path of arbitrary length n given by $\{j_1, \dots, j_n\}$, such that there exists some $c > 0$ with $w_i(i, j_1) \geq c, w_i^{j_{l-1}}(j_{l-1}, j_l) \geq c$ f.a. $l \in \mathbb{N}_{\leq n}$ and $w^{j_n}(j_n, j) \geq c$. Here j is the attacker launching a sybil attack σ_j^n creating n sybil nodes $s_{j1}, \dots, s_{jn}$. This yields a graph $G_i' = G_i \downarrow \sigma_j^n = (V_i', E_i', w_i')$ with $V_i' = V_i \cup \{s_{j1}, \dots, s_{jn}\}$.

Due to independence of disconnected nodes and the fact that no edges have been added yet, the scores of all nodes in V_i have not changed. From i's perspective the nodes $k, s_{j1}, \dots, s_{jn}$ are indistinguishable. Now assume k performs some c units of work for j, then due to misreport-proofness j is best off reporting about this transaction honestly and by several-hop single-report responsiveness it will follow $S_i^M(G_i', k) > S_i^M(G_i, k)$.

Due to symmetry, one could perform any of the isomorphic functions in $\{f_1, \ldots, f_n\}$ where f_l is a graph isomorphism on G_i' which simply swaps the labels of k and s_{jl}. The symmetry requirement implies that any of these graph isomorphisms will yield the same accounting values for all nodes in G_i'. Consequently there exists a report j could make about s_{j1} such that $w_i^j(j, s_{j1}) \geq c$ returning a new graph $G_i^{(1)}$ such that $S_i^M(G_i^{(1)}, s_{j1}) = S_i^M(G_i', k)$. In the next step k may perform some work for s_{j1} and we could apply the graph isomorphism f_2 switching labels k and s_{j2} yielding the graph $G_i^{(2)}$ such that there exists an edge between s_{j2} and s_{j1} with weight $w_i^{s_{j1}}(s_{j1}, s_{j2}) = w_i^j(j, s_{j1})$. By serial-report responsiveness it now follows $S_i^M(G_i^{(2)}, s_{j2}) \geq S_i^M(G_i^{(1)}, s_{j1})$ and $S_i^M(G_i^{(2)}, s_{j1}) \geq S_i^M(G_i^{(1)}, s_{j1})$.

We can continue this for all sybil identities, creating edges with weights $w_i^{s_{jl-1}}(s_{jl-1}, s_{jl}) = w_i^{s_{jl}}(s_{jl}, s_{jl+1})$ as indicated in figure 7.2 and obtain

$$\sum_{l=1}^{n} S_i^M(G_i^{(n)}, s_{jl}) \geq n \cdot S_i^M(G_i^{(1)}, s_{j1}).$$

Due to several-hop single-report responsiveness we know that it now holds $S_i^M(G_i^{(1)}, s_{j1}) > S_i^M(G_i, s_{j1}) = 0$ and therefore $\lim_{n\to\infty} \sum_{l=1}^{n} S_i^M(G_i^{(n)}, s_{jl}) = \infty$.

Lastly, it obviously holds $\omega_-^n(\text{rep}) = S_i^M(G_i^{(1)}, s_{j1}) > 0$. Hence it follows

$$\lim_{n\to\infty} \frac{\omega_+^n(\text{rep})}{\omega_-^n(\text{rep})} \geq \lim_{n\to\infty} \frac{n \cdot S_i^M(G_i^{(1)}, s_{j1})}{S_i^M(G_i^{(1)}, s_{j1})} = \infty.$$

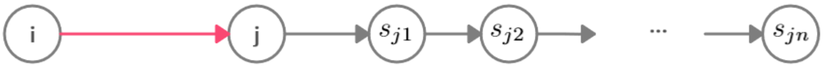


Figure 6.5: Sybil Attack on Serial-report Responsive Accounting Mechanism

□

Analogously, to the corollary, we derived from our theorem about parallel-report responsiveness, we can weaken our definition of serial-report responsiveness, such that it may hold $S_i^M(G_i'', l) \leq S_i(G_i', k)$ and $S_i^M(G_i'', k) \leq S_i(G_i', k)$.

Definition 6.2.6 (Weak Serial-report Responsiveness)**.**
Given a subjective work graph $G_i = (V_i, E_i, w_i)$ of agent i with malicious agent $j \in V_i$ that is not connected to i creating sybil identities $s_{j1}, s_{j2}, \ldots$. We say that an accounting mechanism S_i^M satisfies weak serial-report responsiveness if for a sequence of subjective work graphs $(G_i^{(n)})_{n\in\mathbb{N}}$ with nodes $(s_{jn})_{n\in\mathbb{N}}$ added in each graph with $w_i^{s_{jl-1}}(s_{jl-1}, s_{jl}) \geq c$ and $w_i^j(j, s_{j1}) \geq c$ it holds $\lim_{n\to\infty} \sum_{l=1}^{n} S_i^M(G_i^{(n)}, s_{jl}) = \infty$. Hence, every serial-report responsive accounting mechanism also satisfies weak serial-report responsiveness.

We can derive a rather simple corollary from the upper definition of weak serial-responsiveness.

Corollary 6.2.4.
Every accounting mechanism S^M that satisfies independence of disconnected nodes, symmetry, several-hop single-report responsiveness and weak serial-report responsiveness as well as being misreport-proof has a passive strongly beneficial sybil attack in terms of accounting values.

Proof. The proof is analogous to the one in theorem 6.2.2. □

Analogously to corollary 6.2.2, we can infer that if a strongly beneficial **passive** sybil attack exists for accounting mechanisms satisfying the properties above, then an equivalent active sybil attack with benefit at least as large as above exists.

Corollary 6.2.5.
Every accounting mechanism S^M that satisfies independence of disconnected nodes, symmetry, single-report responsiveness and weak serial-report responsiveness, as well as being misreport-proof, has an active strongly beneficial sybil attack in terms of accounting values.

Note that all of the attacks discussed in the theorems above, were strongly beneficial in terms of accounting values only. If the accounting mechanisms satisfying these properties are, in addition to this *strongly representative*, we conclude that these attacks must be strongly beneficial in terms of work as well.

7

On the Sybil-Proofness of Accounting Mechanisms

In chapter 6 we analysed some important properties that accounting mechanisms must satisfy in order for them **not** to be sybil resistant against strongly beneficial attacks. In this chapter, we would like to do the inverse, namely find properties accounting mechanisms should satisfy in order for them to be resistant to strongly beneficial attacks.

7.1. Characterising Sybil Attacks

We have shown that the properties of parallel-report responsiveness and serial-report responsiveness, in combination with single-report responsiveness, symmetry and independence of disconnected nodes, leads to the existence of strongly beneficial sybil attacks. In terms of accounting values of course. For both of these properties we could conjure a type of attack in an arbitrary work graph which would capitalise on either parallel-report responsiveness or serial-report responsiveness. In the case of parallel-report responsiveness this would be a parallel sybil attack and in the case of serial-report responsiveness it is a serial attack. We will introduce the two below.

Definition 7.1.1 (Parallel Sybil Attack)**.**
Given an arbitrary objective work graph $G = (V, E, w)$ with malicious node $j \in V$. A passive sybil attack of arbitrary size $n \in \mathbb{N}$ given by $\sigma_j^n = (S, E_S, w_S)$ is called a parallel sybil attack if it holds

$$\forall\,(u, v) \in E_S : v \in S,\, u = j.$$

This means that every node in the set of sybils created by the attacker is connected to j, i.e. performs some work for j. Of course, this work is not actually performed, but simply reported to other nodes in the network. The point behind this kind of attack is that all sybil nodes will directly gain from the work j has performed, i.e. the attack edges. An illustration of this kind of attack is given in figure 7.1 below.

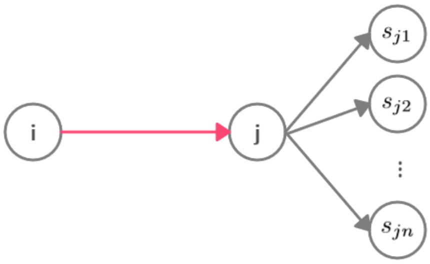


Figure 7.1: Parallel Sybil Attack

Next, we define the concept of a serial attack with the definition 6.2.5 of serial-report responsiveness in mind.

Definition 7.1.2 (Serial Sybil Attack)**.**
Given an arbitrary objective work graph $G = (V, E, w)$ with malicious node $j \in V$. A passive sybil attack of arbitrary size $n \in \mathbb{N}$ given by $\sigma_j^n = (S, E_S, w_S)$ with $S = \left\{s_{j1}, \dots, s_{jn}\right\}$ is called a serial sybil attack if it holds

$$E_S = \left\{(j, s_{j1}), (s_{j1}, s_{j2}), \dots, (s_{jn-1}, s_{jn})\right\}$$

This means that the set of sybil identities is given by a path-like structure in which every sybil is connected to a "predecessor sybil". Visually, this would look like the image given in figure 7.2.

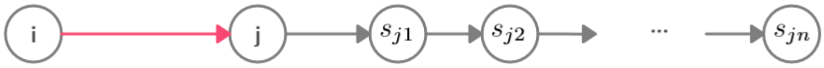


Figure 7.2: Serial Sybil Attack

7.2. Requirements for Sybil-Proofness to Parallel and Serial Attacks

In chapter 6 we introduced two requirements for accounting mechanisms to be susceptible to strongly beneficial parallel- and serial sybil attacks. In this chapter, we will do the inverse, namely introduce requirements for them to be resistant to these types of attacks. We introduce the definitions of convergence of parallel reports and convergence of serial reports.

Definition 7.2.1 (Convergence of parallel reports)**.**
Given a subjective work graph $G_i = (V_i, E_i, w_i)$ of agent i with malicious node $j \in V_i$ such that there exists a path of arbitrary, but finite length $\left\{j_1, \dots, j_n\right\}$ connecting i and j and some $c > 0$ with $w_i^{j_{l-1}}(j_{l-1}, j_l) \geq c$ f.a. $l \leq n$ and $w_i^j(j_n, j), w_i(i, j_1) \geq c$. Now let j perpetrate a parallel sybil attack σ_j^n with sybil identities $\{s_{j1}, \dots, s_{jn}\}$.

Without loss of generality we assume that it holds for the edges $w_i^j(j, s_{jl}) = c_l \leq c_{l-1}$ f.a. $l \leq n$, i.e. we assume non-increasing edge weights, leading to the subjective work graph $G_i^{(n)}$. An accounting mechanism S^M is said to satisfy the **parallel-report bound** if it holds $S_i^M(G_i^{(n)}, s_{jl}) \geq 0$ f.a. $l \leq n$ and for any $n \in \mathbb{N}$ we have

$$\sum_{l=1}^{n} S_i^M(G_i^{(n)}, s_{jl}) \leq S_i^M(G_i^{(1)}, s_{j1}).$$

We now say that the accounting mechanism S^M satisfies **convergence of parallel reports** if it satisfies the parallel-report bound and additionally it holds for any arbitrary sequence $(c_l)_{l\in\mathbb{N}} \subset \mathbb{R}_{\geq 0}$,

$$\lim_{n\to\infty} \sum_{l=1}^{n} S_i^M(G_i^{(n)}, s_{jl}) < \infty.$$

We can now also define an equivalent, albeit slightly relaxed definition for the resistance to serial attacks.

Definition 7.2.2 (Convergence of serial reports)**.**
Given a subjective work graph $G_i = (V_i, E_i, w_i)$ of agent i with malicious node $j \in V_i$ such that there exists a path of arbitrary, but finite length $\{j_1, \ldots, j_n\}$ connecting i and j and some $c > 0$ with $w_i^{j_{l-1}}(j_{l-1}, j_l) \geq c$ f.a. $l \leq n$ and $w_i^j(j_n, j), w_i(i, j_1) \geq c$. Now let j perpetrate a serial sybil attack σ_j^n with sybil identities $\{s_{j1}, \ldots, s_{jn}\}$.

An accounting mechanism is said to satisfy the **serial-report bound** if it holds for any two edge weights $w_i^j(j, s_{j1}) = c_1, w_i^{s_{j1}}(s_{j1}, s_{j2}) = c_2$

$$S_i^M(G^{(n)}, j_n) \leq S_i^M(G^{(1)}, j_1).$$

We now say that an accounting mechanism S^M satisfies **convergence of serial reports** if it holds for some arbitrary sequence $(c_l)_{l\in\mathbb{N}} \subset \mathbb{R}_{\geq 0}$, $S_i^M(G_i^{(n)}, s_{jl}) \geq 0$ f.a. $l \leq n$ with a convergent sum

$$\lim_{n\to\infty} \sum_{l=1}^{n} S_i^M(G_i^{(n)}, s_{jl}) < \infty.$$

From the upper 4 definitions we can derive two rather simple auxiliary lemmas, which we will apply a bit further down the line.

Lemma 7.2.1.
Let G_i be the subjective work graph of honest agent i with attacker $j \in V_i$, launching a parallel sybil attack σ_j^n, according to definition 7.1.1. Let S_i^M be some accounting mechanism of agent i which satisfies convergence of parallel reports according to definition 7.2.1 and path-responsiveness. Then we know already that this attack cannot be strongly beneficial in terms of accounting values.

Proof. The reason this attack cannot be strongly beneficial lies in the fact that due to convergence of parallel reports it must hold $\omega_+^n(\text{rep}) < \infty$ and due to path-responsiveness it must hold $\omega_-^n(\text{rep}) > 0$. This already concludes the proof. □

Lemma 7.2.2.
Let G_i be the subjective work graph of honest agent i with attacker $j \in V_i$ launching a serial sybil attack σ_j^n, according to definition 7.1.2. Let S_i^M be some accounting mechanism of agent i which satisfies convergence of serial reports according to definition 7.2.2 and path-responsiveness. Then we know already that the attack cannot be strongly beneficial in terms of accounting values.

Proof. The proof to this follows the exact same line of reasoning as the proof to lemma 7.2.1. □

7.2.1. Pyramid Sybil Attacks

We believe that making such a strongly highlighted distinction between parallel attacks and serial attacks is done for good reason. We claim that the profit of any passive sybil attack on any arbitrary graph structure can be bounded from above by attacks that are given by the combination of parallel and serial attacks. We refer to the combination of these two as *pyramid attacks*. This will not be as obvious a conclusion as the ones above.

Definition 7.2.3 (Pyramid Sybil Attack)**.**
Given an arbitrary objective work graph $G = (V, E, w)$ with malicious node $j \in V$. A passive sybil attack of arbitrary size $N \in \mathbb{N}$ $\left(N = \sum_{i=1}^{m} n_i\right)$ given by $\sigma_j^N = (S, E_S, w_S)$ with $S = \left\{s_{j11}, s_{j12}, \ldots, s_{j1n_1}, s_{j21} \ldots, s_{j2n_2} \ldots, s_{jm1}, \ldots, s_{jmn_m}\right\}$ is called a pyramid sybil attack if it holds

$$\forall (j, u) \in E_S : u \in \left\{s_{j11}, \ldots, s_{j1n_1}\right\}$$

and

$$\forall 1 < l \leq m \forall i \leq n_l \exists! k \leq n_{l-1} : (s_{l-1k}, s_{li}) \in E_S.$$

Visually, this type of attack looks as follows

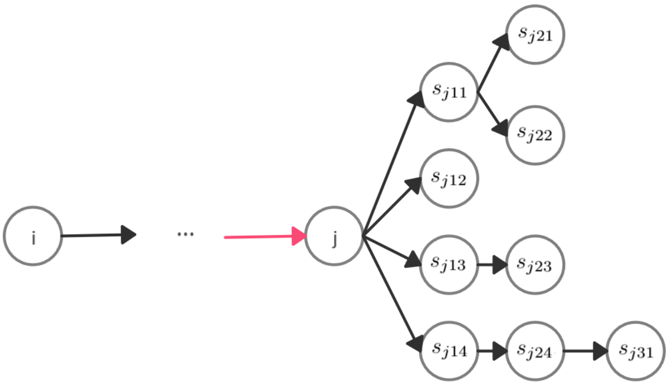


Figure 7.3: Pyramid Sybil Attack

This type of sybil attack is given by a set of layers. In each layer the sybil attack can be interpreted as given by a number of parallel sybil attacks. This means that every sybil identity created by the attacker performs some counterfeit work for exactly one other sybil identity, which is located one layer above in the sybil region. The point here is that this type of attack can be interpreted as a combination of serial and parallel attacks. The branches of the pyramid are serial attacks and the layers are parallel attacks. Our goal now is to combine the properties given in definitions 7.2.1 and 7.2.2 to make an accounting mechanism resistant to the more generic pyramid attack. Note that parallel and serial sybil attacks are special cases of pyramid attacks and therefore any accounting mechanism that is resistant to pyramid attacks is also resistant to parallel and serial attacks.

Proposition 7.2.1.
Let G_i be the subjective work graph of honest agent i with attacker $j \in V_i$ launching a pyramid sybil attack σ_j^N, according to definition 7.2.3 of variable size $N = \sum_{i=1}^{m} n_i$. Let S_i^M be some accounting mechanism of agent i which satisfies convergence of serial reports, convergence of parallel reports and path-responsiveness. Then we know already that the attack cannot be strongly beneficial in terms of accounting mechanisms.

Proof. The proof of this theorem follows from the fact that every pyramid sybil attack is nothing, but a combination of parallel and serial sybil attacks. We begin by only examining the first layer of the pyramid attack, i.e. $\left\{s_{j11}, \ldots, s_{j1n_1}\right\}$. The given pyramid confined to this layer is a simple parallel sybil attack and we know by the convergence of parallel report property that it must hold

$$\lim_{n_1 \to \infty} \sum_{k=1}^{n_1} S^M(G_i', s_{j1k}) < \infty.$$

This means that the attacker cannot gain infinite accounting values in the first layer by only scaling this first layer of their sybil attack. Now, the second layer of the pyramid attack can be interpreted as a number of sybil attacks perpetrated by n_1 attackers. By the parallel-report bound property, we already know that the profit of each of these layer attacks must be bounded by

$$\max\left\{S_i^M(G_i, s_{j21}), \dots, S_i^M(G_i, s_{j2n_2})\right\},$$

and that the attacker cannot increase this to any arbitrarily large value by scaling the first layer. This is because the convergence of parallel reports dictates that it then follows $S_i^M(G_i, s_{j1l}) \to 0$ for all but finitely many $l \in \mathbb{N}$ and by the serial-report bound property it must then also follow $S_i^M(G_i, s_{j2l}) \to 0$ for all $l \leq n_2$ for which s_{j2l} is connected to a node for which it holds $S_i^M(G_i, s_{j1l}) \to 0$. Hence scaling layer 1 will yield

$$\lim_{n_1 \to \infty} \sum_{l=1}^{n_2} S_i^M(G_i', s_{j2l}) < \infty$$

and consequently

$$\lim_{n_1 \to \infty} \sum_{l=1}^{n_1} S_i^M(G_i, s_{j1l}) + \sum_{l=1}^{n_2} S_i^M(G_i, s_{j2l}) < \infty.$$

By logic of mathematical induction we can conclude that scaling any layer $k \leq m$ of the sybil attack will always yield a finite aggregate of accounting values, i.e.

$$\lim_{n_{k-1} \to \infty} \sum_{l=1}^{n_1} S_i^M(G_i, s_{j1l}) + \sum_{l=1}^{n_2} S_i^M(G_i, s_{j2l}) + \ldots + \sum_{l=1}^{n_k} S_i^M(G_i, s_{jkl}) < \infty.$$

Hence we know that scaling any layer of the pyramid sybil attack will result in all following layers returning accounting values of 0 for all but finitely many sybils.

So far we have established that a pyramid sybil attack cannot be made to yield an infinite sum of accounting values by scaling any set of layers to infinity, due to convergence of parallel reports and bounded transitive trust. Now, the only other possible alternative for attempting this is by scaling the number of layers of the attack, i.e. $m \to \infty$, where each layer will contain finitely many sybils ($n_k < \infty$ f.a. $k \leq m$).

Due to the parallel-report bound property this would mean that the profit of this type of scaling would then be bounded from above by the profit of an infinite serial sybil attack $G_i'' = G_i \downarrow \tilde{\sigma}_j^n$ with sybil nodes $\{j, s_{j1}, s_{j2}, \ldots\}$, where each s_{jk} is given by

$$s_{jk} = \arg\max\left\{S_i^M(G_i, s_{jkl}) \mid l \leq n_k\right\}.$$

However, by convergence serial reports we know that it must follow

$$\sum_{l=1}^{\infty} S_i^M(G_i, s_{jl}) < \infty.$$

Therefore we know that for a pyramid attack of arbitrary size it must hold $\omega_+^n(\text{rep}) < \infty$, while by path-responsiveness we conclude that $\omega_-^n(\text{rep}) > 0$, which concludes our proof. □

7.3. Bounding the Profit of Arbitrary Passive Sybil Attacks

At this point we know that all accounting mechanisms satisfying the properties discussed above are resistant to strongly beneficial pyramid attacks. Next, we introduce one additional property accounting mechanisms must satisfy in order for the profit of any passive sybil attack to be bounded by the profit of an arbitrary, but fixed number of pyramid attacks. We call this property *multiple-path response bound.*

Definition 7.3.1 (Multiple-Path Response Bound)**.**
Let G_i be the subjective work graph of honest node i containing node $k \in V_i$ such that there exist N paths $(P_n)_{n \leq N}$ connecting k to i. Now, define G_i' as an altered version of the subjective work graph of i, whereby the agent k is "split" into several agents $k_1, \ldots, k_N$, where every k_l $(l \leq N)$ is connected to i by exactly one path.

G_i' is created by splitting the node k into as many nodes as there are paths connecting to it. We begin with k_1 and remove all nodes and edges that are part of any of the paths $P_2, \ldots, P_N$ while keeping all which are part of P_1. This means, we only keep edges and nodes that are either part of the first path, or that are not in any of the paths at all. We now relabel k (as the end-point of P_1), k_1. Next, we add path P_2 to the graph whereby we remove all edges and nodes that are part of any paths $P_3, \ldots P_N$. Any node j (or edge e) in P_2 that is also part of P_1, is now duplicated into j_1 and j_2 such that $j_1 \in P_1$ and $j_2 \in P_2$, i.e. $(e_1 \in P_1$ and $e_2 \in P_2)$. We continue this for all paths $P_1, \ldots, P_N$ and obtain G_i'.

Then we say that the accounting mechanism S^M satisfies the multiple-path response bound if it holds

$$S_i^M(G_i, k) \leq \sum_{n=1}^{N} S_i^M(G_i', k_n).$$

Given below, in figures 7.4, 7.5 and 7.6 are some examples of graphs G_i and G_i' illustrating the multiple-path response bound

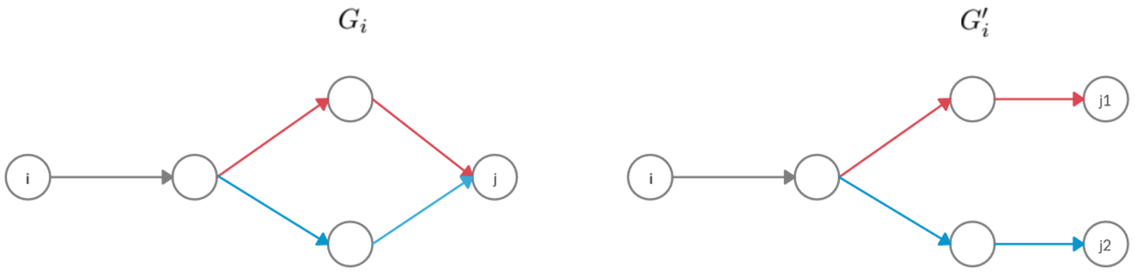


Figure 7.4: Example of Multiple-path Response Bound

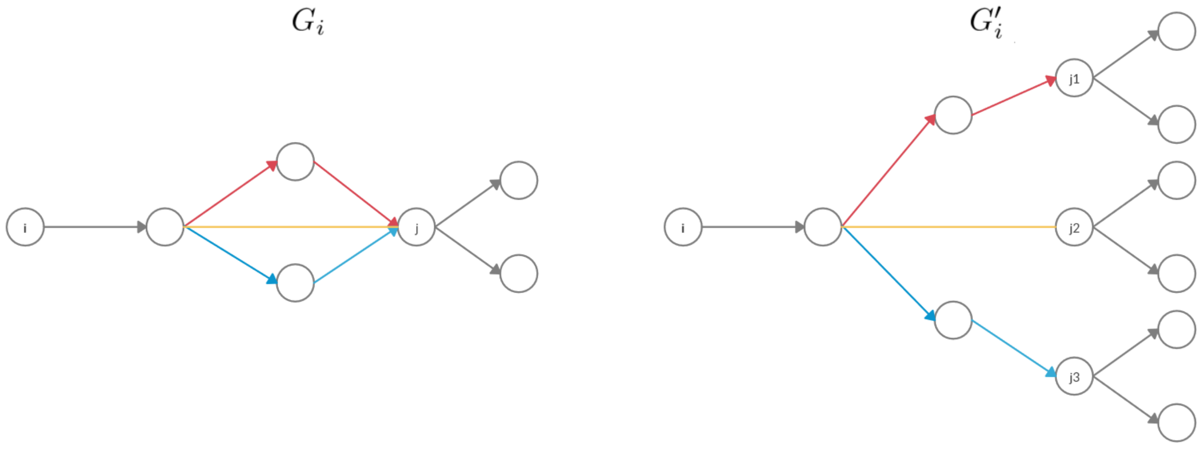


Figure 7.5: Example of another Multiple-path Response Bound

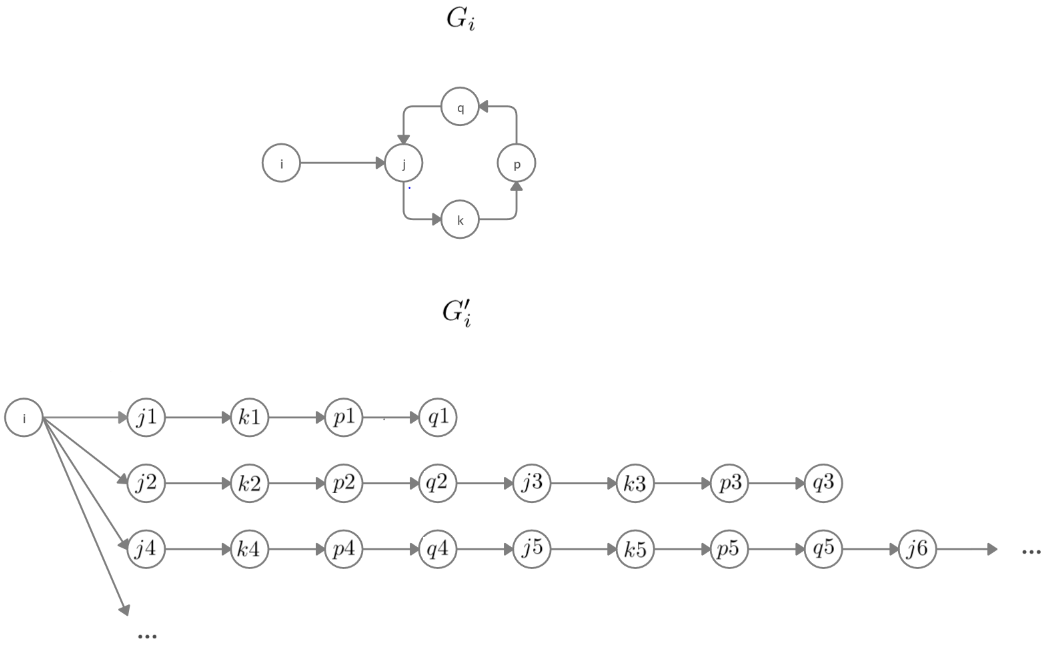


Figure 7.6: Example of Multiple-path Response Bound with Loops

This may at first seem like a rather restrictive assumption that eliminates many of the common accounting mechanisms. However, we find that this is in fact not true. The upper definition is satisfied by all accounting mechanisms defined in [29]. Instead, we provide an intuition for why this actually makes sense.

In order for an accounting mechanism to determine the trustworthiness of another node, it needs to evaluate this node's contributions and leeches to and from the network. As there is always the possibility of faking edges, we want to limit the effect that edges which are in between unknown nodes have on the accounting values, and only take into account the contributions made to nodes that i has at least an indirect connection to. This means we want to evaluate each node j by the incoming edges from i. Each path connecting j to i can be considered an indirect contribution and therefore should influence the accounting value of j in i's subjective work graph.

However, it is crucial that the effect of an additional path in the network should not exceed the effect that this additional path would have on $S_i^M(G_i, j)$ if it were the only path connecting j to i. We feel like this is a fairly intuitive and sensible definition, which is satisfied by plenty of the existing accounting mechanisms such as PageRank, Maxflow and Netflow.

The reason we introduce the definition of multiple-path response bound is that we can now bound the profit of every passive sybil attack from above by the profit of an equivalent pyramid sybil attack.

In order to achieve this goal we introduce the additional definitions of transitive trust and bounded transitive trust.

Definition 7.3.2 (Transitive Trust)**.**
Let G_i be the subjective work graph of node i containing nodes $j, k \in V_i$, such that $w_i(i, j), w_i(j, k) > 0$. Then we say that an accounting mechanism S_i^M satisfies the **transitive trust** property if it holds

$$S_i^M(G_i, j) > 0 \,\&\, S_j^M(G_j, k) > 0 \Rightarrow S_i^M(G_i, k) > 0.$$

In line with earlier extensions to multiple hops we extend this to multiple hops as well and state that the accounting mechanism satisfies **several-hop transitive trust** if for a path $(i, j_1, \ldots, j_n, j)$ of fixed, but arbitrary length n with $w_i(i, j_1) > 0$, $w_i(j_{l-1}, j_l) > 0$ f.a. $l \leq n$ and $w_i(j_n, j) > 0$ it holds

$$S_i^M(G_i, j_1) > 0, S_{j_l}^M(G_{j_l}, j_{l+1}) > 0, \ldots, S_{j_n}^M(G_{j_n}, j) > 0 \Rightarrow S_i^M(G_i, j) > 0.$$

Definition 7.3.3 (Bounded Transitive Trust)**.**
Let G_i be the subjective work graph of node i containing nodes $j, k \in V_i$, such that $w_i(i, j), w_i(j, k) > 0$. We now say that S^M satisfies **bounded transitive trust** if it satisfies transitive trust and it additionally holds

$$S_i^M(G_i, k) \leq \min\left\{S_i^M(G_i, j), S_j^M(G_j, k)\right\}.$$

Extended to several hops it should hold for a path $(i, j_1, \ldots, j_n, j)$ of fixed, but arbitrary length n with $w_i(i, j_1) > 0$, $w_i(j_{l-1}, j_l) > 0$ f.a. $l \leq n$ and $w_i(j_n, j) > 0$

$$S_i^M(G_i, j) \leq \min\left\{S_i^M(G_i, j_1), S_{j_1}^M(G_{j_1}, j_2), \ldots, S_{j_n}^M(G_{j_n}, j)\right\}.$$

Note that if there are several paths connecting i and k then $S_i^M(G_i, k)$ must be bounded by the sum of the minimums given above (for each path). So if there are N paths $(P_n)_{n \leq N}$ connecting i and k we should obtain the following upper bound.

$$\sum_{n=1}^{N} \min\{S_{j_l}(G_{j_l}, j_{l+1}) \mid j_l \in P_n\}.$$

The definition of transitive trust describes the concept that if a node i trusts another node j, and j trusts another node k then it must already follow that i has some trust in node k as well, while bounded transitive trust implies that the trust i has in k must be bounded from above by both the trust i has in j and the trust that j has in k.

We can now prove the lemma below.

Lemma 7.3.1.
Let S_i^M be an accounting mechanism satisfying path-responsiveness, the multiple-path response bound and bounded transitive trust. Now let G_i be the subjective work graph of honest agent i with $|V_i| < \infty$. Let j be a malicious agent launching a passive sybil attack σ_j^n on G_i such that there exist one or more paths connecting j to i. Then the profit $\omega_+^n(\text{rep})$ is bounded by the profit $\tilde{\omega}_+^n(\text{rep})$ of an equivalent passive pyramid sybil attack mutiplied by a constant $c < \infty$.

Proof. Let's start off with the simple case of the passive sybil attack by j which is connected to i, where we assume that there is only one path connecting i to j. In this case the multiple-path response bound makes the statement above trivially correct, we can simply restructure the sybil region in such a way that every sybil is connected to j via a single path. Naturally, we obtain a pyramid sybil attack.

Now assume that there are finitely many (n) paths connecting i and j then according to the multiple-path response bound, we can restructure the subjective work graph G_i such that we obtain $j_1, \ldots, j_n$, each connected to i via a single path and all perpetrating the same sybil attack. In the next step we apply the multiple-path response bound property again and obtain finitely many pyramid sybil attacks. We then obtain the upper bound for the sybil attack profit

$$n \cdot \tilde{\omega}_+^n(\text{rep})$$

where $\tilde{\omega}_{+}^{n}(\text{rep})$ is the largest profit of any of the n pyramid sybil attacks.

Lastly, if there are infinite paths connecting i and j then we know by $|V_i| < \infty$ that at least one of these paths must contain a loop. Now, the bounded transitive trust property ensures that the accounting values of any nodes that 'follow' the loop are not larger than any the accounting values of nodes that came before the loop and we can therefore without loss of generality remove all loops from the subjective work graph, without affecting the sybil attack profit. We arrive at the same conclusion as we did for sybil attacks with finitely many paths between j and i. □

7.4. Final Results on Sybil-proofness

We now have obtained a pretty strong result about the sybil-proofness of accounting mechanisms against pyramid attacks as well as a result about the fact that the profit of every passive sybil attack can be bounded by the multiple of the profit of a pyramid sybil attack. This leads to what we believe is a rather strong theorem on sybil resistance.

Theorem 7.4.1.
Any accounting mechanism S^M satisfying path-responsiveness, multiple-path response bound, convergence of serial reports and convergence of parallel reports as well as bounded transitive trust on a finite subjective work graph G_i is resistant to strongly beneficial passive sybil attacks.

Proof. Let $G_i = (V_i, E_i, w_i)$ be the subjective work graph of agent i with $|V_i| < \infty$. Let j be a malicious node launching a passive sybil attack σ_j^n of arbitrary size $n \in \mathbb{N}$. Then due to the bounded transitive trust property we can without loss of generality assume that there are finite paths connecting i and j. By mutiple-path responsiveness we know that the profit of the sybil attack is bounded from above by several pyramid attacks of equal size, each connected to i by exactly one path. Now by convergence of serial reports and convergence of parallel reports we know that all of these pyramid attacks yield a bounded profit. Hence, we find that $\omega_{+}^{n}(\text{rep}) < \infty$ and due to path-responsiveness we know that $\omega_{-}^{n}(\text{rep}) > 0$. This already concludes our theorem.

□

We now extend this result to active sybil attacks and bound the profit of these by the same logic as in the theorems above.

Corollary 7.4.1.
Any accounting mechanism S^M satisfying path-responsiveness, multiple-path response bound, convergence of serial reports and convergence of parallel reports as well as bounded transitive trust is resistant to strongly beneficial active sybil attacks.

Proof. In the case of an active sybil attacks there exist attack edges which connect to sybil agents in S. We can assume here that there are finitely many of these. By the multiple-path response bound we know that the profit of an active sybil attack is bounded by the the profit of a finite number of (passive) pyramid sybil attacks, each perpetrated by the sybil nodes in with attack edges connected to them. By the same multiple-path response bound we also know that the profit of each of these pyramid attacks is bounded through the number of paths connecting them to i. Both of these are finite as the subjective work graph G_i is finite.

The rest of the proof to this follows simply from theorem 7.4.1 as we have obtained finitely many pyramid attacks. As before, the convergence of serial reports as well as the convergence of parallel reports and the bounded transitive trust property return a finite profit. Path-reponsiveness implies a sybil attack cost > 0 which yields $\lim_{n\to\infty} \frac{\omega_{+}^{n}(\text{rep})}{\omega_{-}^{n}(\text{rep})} < \infty$ as before.

□

We can conclude this section by saying that any accounting mechanism that satisfies the requirements from theorem 7.4.1 is resistant to strongly beneficial sybil attacks in terms of accounting mechanism. Any accounting mechanism that additionally satisfies at least weak representativeness will then also be resistant in terms of the amount of work that sybils can consume.

The question now arises which accounting mechanisms actually satisfy these requirements and whether we can find such an accounting mechanism which is also at least weakly representative. A particular example of such an accounting mechanism is given in example 7.4.1 below.

Example 7.4.1.
Let G_i be the subjective work graph of agent i and S_i^{PHT} be the personalised hitting time algorithm as introduced in [11], i.e. let $(X_0, X_1, \ldots, X_\tau)$ be an α-terminating random walk on the subjective work graph G_i, where each $X_i \in V_i$ and

$$\mathbb{P}\left(X_{t+1} = j \mid X_t = i\right) = (1-\alpha) \cdot \frac{w(i,j)}{\sum\limits_{(i,j') \in E_i} w(i,j')},$$

and the walk length is a random variable $\tau \sim Geom(1-\alpha)$.

Then the personalised hitting time values of agent $j \in V_i$ is given by

$$S_i^{\mathrm{PHT}}(G_i, j) = \mathbb{P}\left(j \in (X_t)_{t=0}^{\tau} \mid X_0 = i\right).$$

The accounting mechanism $S_i^{\mathrm{PHT}}(G_i, j)$ then satisfies all of the requirements for theorem 7.4.1 to hold, i.e. parallel-report bound, convergence of parallel reports, convergence of serial reports, bounded transitive trust and multiple-path response bound.

By theorem 7.4.1 we therefore know that S^{PHT} is resistant to strongly beneficial sybil attacks in terms of accounting values. However, the question arises whether it is also resistant to strongly beneficial sybil attacks in terms of work. In the example below, we will show that this is not the case, i.e. S^{PHT} does not satisfy weak representativeness.

Let G_i be a subjective work graph of agent i containing honest agent k and attacker j, creating a parallel sybil attack consisting of 3 sybils s_{j1}, s_{j2}, s_{j3}, each connected to j by an edge of variable weight N. We set $w_i(i,j) = 9$ and $w_i(i,k) = 1$. Now, the hitting time algorithm with $\alpha = 0.1$ returns the values given in figure 7.7 below.

$S_i^M(G_i,k)$	$S_i^M(G_i,j)$	$S_i^M(G_i,s_{j1})$	$S_i^M(G_i,s_{j2})$	$S_i^M(G_i,s_{j3})$
0.1	0.9	0.27	0.27	0.27

Figure 7.7: Sybil attack profit in terms of accounting values (1)

Now, if k and s_{j1} query i for some data s_{j1} will be served and we obtain a new subjective work graph G_i' containing the edge $w(S_{j1}, i) = 1$. We now recompute the accounting values for all nodes in this new graph and obtain the values given in figure 7.8 below.

$S_i^M(G_i', k)$	$S_i^M(G_i', j)$	$S_i^M(G_i', s_{j1})$	$S_i^M(G_i', s_{j2})$	$S_i^M(G_i', s_{j3})$
0.12187	0.9	0.27	0.33	0.33

Figure 7.8: Sybil attack profit in terms of accounting values (2)

Hence, after leeching from i, s_{j1} still has a higher accounting value than k and therefore s_{j1} can leech infinitely from i, which means the attack given in figure 7.9 is weakly beneficial in terms of accounting values, but strongly beneficial in terms of work. This means the existing personalised hitting time accounting mechanism is not weakly representative.

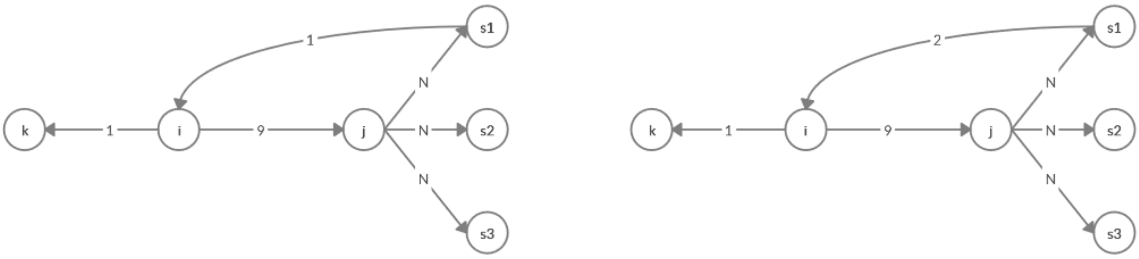


Figure 7.9: Strongly Beneficial Sybill Attack in terms of work on PHT

We now make S^{PHT} weakly representative by introducing an additional constraint, by choosing accounting mechanism

$$S_i^M(G_i, j) = \max\left\{\sum_{k\in V_i} w_i(i,k) - w_i(k,i), 0\right\}\cdot S_i^{\mathrm{PHT}}(G_i, j).$$

This accounting mechanism satisfies all requirements from theorem 7.4.1 as well as weak representativeness as any node in V_i with finite accounting values can only leech finite amounts of data from a given node. In fact, it even satisfies strong representativeness as an agent j leeching infinite amounts of work from a another agent i must imply that agent j has received infinite work. The only way this could have happened is if the attacking agent has made this infinite contribution.

8
Conclusion and Discussion

In this thesis we have examined reputation mechanisms in distributed systems and their resistance to different types of malicious behaviour, whereby we placed the largest emphasis on sybil attacks. We began by introducing a mathematical framework for our research in which we defined fundamental and pertinent concepts such as transaction sequences, work graphs, accounting mechanisms and allocation policies. Thereafter we mathematised different types of malicious behaviour, accounting mechanisms were supposed to prevent, i.e. lazy-freeriding, misreport attacks and sybil attacks. It was our goal to introduce requirements for accounting mechanisms to be resistant to this type of malicious behaviour.

We began by investigating **lazy-freeriding** in chapter 3. We discovered a combination of requirements that would ensure an accounting mechanism together with an allocation policy could successfully prevent lazy-freeriding. The requirement was called positive-report responsiveness, while the allocation policy had to satisfy the additional constraint of banning any nodes from the choice set with accounting values that exceeded a given lower bound. This resulted in agents, who contributed far fewer resources than they consumed, not being served anymore data by honest agents. Therefore it became impossible for agents to leech excessively.

Next we analysed **misreports** and the resistance of accounting mechanisms to these types of attacks in the network, whereby we began by critically examining the DropEdge protocol introduced by Seuken & Parkes [26]. We discovered that this mechanism was only resistant to particular types of misreports which were quite narrowly defined. In response to this discovery we expanded our definition of misreport-proofness and examined the TrustChain data structure in combination with generic gossip protocol. We concluded that TrustChain satisfied a stronger requirement for misreport-proofness for accounting mechanisms, provided that accounting mechanisms satisfied the property of positive-report responsiveness. Given this misreport-proofness we moved on to the most critical issue accounting mechanisms faced, namely sybil attacks.

The largest emphasis was placed on **sybil attacks** in this thesis. After having solved the issue of misreports by either the DropEdge mechanism or the TrustChain architecture we moved on to characterising the effects of sybil attacks on P2P networks. We began by defining the cost incurred by the attacker, given by the amount of work that had to be performed for the network and formalised the profit of an attack as well, whereby the profit of a sybil attack was given by the additional amount of work the attacker could consume after the sybil attack had been carried out. The fact that neither profit nor cost of sybil attacks have been rigorously defined up until this point proves itself to be quite problematic and we highlighted the necessity for these definitions. In attempting to determine the profit of such an attack, we realised that we had to determine the expected value of an infinite discrete stochastic process. To solve this problem we postulated an interaction model for nodes in the network.

Interaction Model

In order to determine the profit of a sybil attack, we introduced an interaction model by which participants transact with one another and compared the outcome of this interaction model with the real-world *Tribler* application. Within the construction of this model we analysed a number of different allocation policies for their resistance to sybil attacks and decided that the winner-takes-all policy was the most suitable in this endeavour. With the now won definiton of sybil attack profit, we realised that computing the value of the expected profit was practically impossible. This prompted us to reformulate the values of cost and profit in terms of the accounting values that sybil attackers were able to obtain. The advantage to this was that the profit in terms of accounting values was, in fact, computable and we were therefore able to gauge the effectiveness of a sybil attack. We did, however incur a problem with this definition.

Representativeness

The values of sybil attack cost and profit in terms of accounting values were now much easier to compute. However, they were not actually the relevant metric, but just a proxy for the earlier defined cost and profit of sybil attacks in terms of work. Consequently, we were interested in determining the relationship between the two and came up with two example cases in which the sybil attack benefit converges to infinity in terms of one, but not the other. We learned that the two were not equivalent. This lead to the new definitions of weak and strong *representativeness*. After having introduced representativeness, we concluded that any accounting mechanism that is resistant to strongly beneficial sybil attacks in terms of accounting values and is weakly representative, is already resistant to strongly beneficial sybil attacks in terms of work, which was the desired property.

Impossibility Results

Next, we analysed some existing results in the literature which stated that accounting mechnanisms satisfying independence of disconnected agents, symmetry and single-report responsiveness were **not** resistant to strongly beneficial sybil attacks in terms of work. We concluded that given our definitions of sybil attack cost and profit this result was incorrect. We corrected the result by adding the requirement of parallel-report responsiveness and added that the consequent sybil susceptibility was in terms of accounting values and not work. If an accounting mechanism satisfied the additional property of strong representativeness then the given sybil susceptibility was also in terms of work. We then extended the model to multiple hops ensuring that the upper assertions were true for arbitrary work graphs. We then introduced another property called serial-report responsiveness and made an analogous assertion.

Sybil-proofing Accounting Mechanisms

Using the impossibility results we arrived at in chapter 6, we realised we could characterise passive sybil attacks as parallel and serial attacks and a combination of the two, which we called pyramid attacks. We inverted the concepts of parallel- and serial-report responsiveness to obtain resistance against these types of attacks. Thereafter we introduced the very important definition of multiple-path response bound from which we concluded that the profit in terms of accounting values of any sybil attack could be bounded by the profit of some pyramid attack, multiplied by a constant. This lead to the final result that any accounting mechanism satisfying the upper properties was resistant against strongly beneficial sybil attacks in terms of accounting values, which in combination with weak representativeness implied resistance to strongly beneficial sybil attacks in terms of work.

Given these results we believe that we have adequately answered the research question from chapter 1

What requirements does an accounting mechanism need to satisfy in order to effectively incentivise cooperation and prevent lazy freeriding, while being resistant to misreport attacks and mitigating the effects of sybil attacks?

8.1. Future Work

In this thesis we covered a wide array of problems in the context of cooperation in P2P file sharing networks. We are particularly pleased with some of our results on the theoretical properties accounting mechanisms and allocation policies must satisfy in order to achieve sybil-proofness. Given the time constraints of a master thesis project, some of our research was cut short a bit. In this section we would like to elaborate a little bit on possible future work that could be conducted in researching the problems this thesis covered.

Allocation Policies

The research on allocation policies we conducted within defining the sybil attack profit in terms of work, has been rather slim and did not reach a final strong conclusion. We determined that out of the allocation policies introduced in chapter 3 the winner-takes all was the most sybil resistant and lead to the fairest distribution of data in the network as was seen in the experiments conducted in 4.1.3. In future work, one may want to formalise a more generic set of sensible allocation policies and determine the optimal policy out of this set. One may determine a set of allocation policies given by convex combinations of the top n policies, i.e. combinations of distribution and top n policies where the highest ranking n nodes will be served each with an amount of work corresponding to their standing in the choice set. If combined with some banning element, whereby only nodes with accounting values greater than a given upper bound are served, such allocation policies may succeed in preventing both lazy freeriding and sybil attacks. Research on allocation policies to our knowledge has been rather scarce and may be a topic of research that hides promising results.

Resistance Against Weakly Beneficial Sybil Attacks

The research conducted for this thesis was all done with P2P file sharing networks in mind, more precisely the *Tribler* application. In this setting weakly beneficial sybil attacks are not by any stretch of the imagination disastrous for the network. The reason for this is that in filesharing networks a weakly beneficial sybil attack in terms of work requires the attacker to invest infinite resources in order obtain infinite resources. This renders weakly beneficial sybil attacks comparatively harmless. No single malicious agent can simply demand all, or even a significant proportion of the resources in the network. However, other types of P2P networks such as social networks may not have such resistant properties. If one thinks of a sybil attacker on Facebook that aims to spread fake news by tricking the network into considering their content more relevant for people's feeds than it actually is. In such a case, a weakly beneficial sybil attack may have disastrous consequences. For future work it may be very nice to obtain some stricter finite upper bounds on the benefits of sybil attacks. For this, one could start of by tightening the definitions of parallel- and serial-report convergence to obtain a limit $\leq c$ for some $c > 0$.

Expanding on Representativeness of Accounting Mechanisms

In chapter 5 we introduced the concept of representativeness of accounting mechanisms. The idea was that accounting values were simply a representation of the reputation of nodes in the network and therefore a proxy for the amount of work these nodes were entitled to. We incurred the problem of sybil attacks which were strongly beneficial in terms of accounting values but not in terms of work, and vice versa, prompting us to make the restriction for accounting mechanisms to have to satisfy at least weak representativeness in order to be sensible. In remark 5.2.1 to further elaborate on what requirements accounting mechanisms must satisfy in order to be weakly and strongly representative we explained the concept of a representativeness function, but did not delve further into this issue. It is, however a crucial point in the resistance to sybil attacks. In future work one may want to further identify properties representative accounting mechanisms satisfy and make consequent additional restrictions to be able to better identify sensible accounting mechanisms.

8.2. Further Discussion and Ethical Ramifications

The research question of this thesis has far-reaching implications and may find application in many types of networks other than just P2P filesharing networks. Digital currencies may be one of these, where the accounting values would not necessarily reflect an agent's trustworthiness, but instead the balance of their account, i.e. the amount of digital money they own. Another preeminent setting for accounting mechanisms to find application in, are online social networks. Networks such as Facebook and Twitter struggle to clamp down on the spreading of hateful speech and fake news. Malicious agents may decide to create many fake accounts which all "follow" or "like" one another in order to boost the probability of their content being seen by many people. The algorithms of these companies that decide whose content is recommended and shown on other honest agents' feeds so far have been very susceptible to these types of attacks. Online social networks try to shut down attacks like this with the help of machine learning algorithms that are trained to detect fake accounts. However, oftentimes mistakes are made and anyone who has used these services has witnessed this first hand. Companies running these services may want to consider broadening their set of tools with which they tackle these attacks by sybil-proof accounting mechanisms that represent the trustworthiness of agents in the network.

One more application that has occured to us throughout the process of this research have been real-life social scoring systems. Countries like China have introduced social credit systems with which they aim to rate their citizens trustworthiness. Bad behaviour of citizens will lead to lower scores while good behaviour will increase respective scores [2]. These scores are then used by the government to allow their citizens different levels of liberty, whereby lower ranking citizens are restricted in their freedom. From a western libertarian perspective this extent of government surveillance seems obviously unethical (of course this is debatable). We are aware of the fact that our research in the direction of social reputation scores may assist oppresive regimes in constructing social accounting mechanisms to enhance their control over peoples' lives. Of course, we realise that our research is only very peripherally related, nevertheless we feel that this must be pointed out and further research in this direction should be conducted with some level of caution and awareness of its ethical consequences.

A

Appendix

A.1. Reputation Dynamics in Indirect Reciprocity

A.1.1. Axioms defining Reputation Mechanism

In order for us to determine what properties a reputation mechanism needs to satisfy we looked into evolutionary biology. In [17] the concept of a binary honour score coupled with a set of behavioural strategies is introduced. An honour score is based not on a node's entire transaction history, but on its most recent transaction. The set of honour scores is given by $\{0,1\}$ and the set of strategies by $\{C,D\}$, whereby C stands for cooperate and D for defect. It has been shown that conditional and unconditional altruism leads to cooperation among a population. Every node has a reputation value of either 0 or 1 and every node has a behavioral strategy, given by

$$p : \{0,1\}^2 \to \{C,D\}.$$

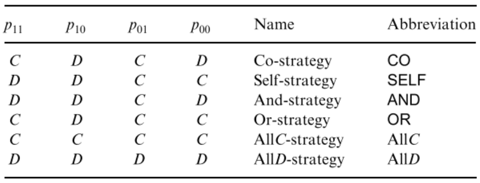
p determines whether a node with a given reputation value will cooperate with a node of another reputation value. For instance $p(0,1) = C$ means that a node with reputation value 0 will cooperate with a node of reputation value 1. A reputation dynamic is a function that assigns a node that has made a decision whether to cooperate or defect with another node a new reputation value, i.e.

$$d : \{0,1\}^2 \times \{C,D\} \to \{0,1\}.$$

In this case for instance $d(0,1,C) = 1$ implies that if a node of reputation 0 cooperates with a node of reputation 1 then it will be assigned reputation 1.

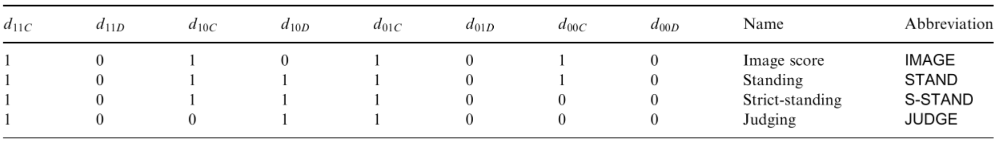
This yields 2^8 possible reputation dynamics and 2^4 behavioural strategies. Note that a reputation dynamic is fixed and population dependent whereas a behavioural strategy is personal and node-specific.

Ohtsuki et al. identify a number of beahvioural strategies and reputation dynamics of particular importance [17].



p_{11}	p_{10}	p_{01}	p_{00}	Name	Abbreviation
C	D	C	D	Co-strategy	CO
D	D	C	C	Self-strategy	SELF
D	D	C	D	And-strategy	AND
C	D	C	C	Or-strategy	OR
C	C	C	C	All$C$-strategy	All$C$
D	D	D	D	All$D$-strategy	All$D$

Figure A.1: Behavioural Strategies (taken from [18]).



d_{11C}	d_{11D}	d_{10C}	d_{10D}	d_{01C}	d_{01D}	d_{00C}	d_{00D}	Name	Abbreviation
1	0	1	0	1	0	1	0	Image score	IMAGE
1	0	1	1	1	0	1	0	Standing	STAND
1	0	1	1	1	0	0	0	Strict-standing	S-STAND
1	0	0	1	1	0	0	0	Judging	JUDGE

Figure A.2: Reputation Dynamics (taken from[18]).

A.1.2. Requirements for Reputation Mechanisms

A combination of reputation dynamic and behavioural strategy (p, d) is called evolutionary-stable strategy (ESS) if p is evolutionarily stable among all 16 possbile behavioural strategies given the reputation dynamic d. This means that given the reputation dynamic d the behavioural strategy p receives on average the highest payoff among all other 15 behavioural strategies, given that the population is dominated by agents with the same behavioural strategy, i.e. more than 50% of the network exhbibit the same strategy. As a corollary statement this implies that the benefit of increasing one's reputation must exceed the cost of the work performed for this increase in reputation. An ESS is a refined form of a Nash equilibrium.

We assume that participating in a P2P-filesharing network constitutes a multi-player game-theoretical game, given by $\mathcal{G}_n = (S, d, E)$ where n is the number of participants, $S = \left\{p_i : \{0,1\}^2 \to \{C, D\} | 1 \leq i \leq n\right\}$ is the set of behavioural strategies of all agents in the network. $E(p, S)$ is then the expected payoff or profit of an agent with behavioral strategy p in the network $\mathcal{G}_n$. Note that this payoff function is a stochastic expected value and not deterministic, because it depends on who the agent interacts with. This expected payoff is given by $\mathbb{E}[b(p_X(r(X), r(i))] - \mathbb{E}[c(p_i(r(i), r(X))]$, whereby X is a random variable choosing players in the network with a predetermined probability distribution f_X, b and c correspond to the benefit and the cost of a possible cooperation and/or defection.

Definition A.1.1 (ESS Strategy)**.** The expected payoff of a player with strategy p in a network where l of the $n-1$ remaining nodes play with the strategy p and $n-1-l$ of the players play with strategy q, can be written $E(p, p^l, q^{n-1-l})$. A strategy p is said to be evolutionarily stable with respect to another strategy q if there exists a $j \in \{1, \dots, n-1\}$ such that

$$\forall i \leq j : \; E(p, p^{n-1-i}, q^i) \geq E(q, p^{n-1-i}, q^i) \tag{A.1}$$

$$\forall i > j : \; E(p, p^{n-i-1}, q^i) > E(q, p^{n-i-1}, q^i). \tag{A.2}$$

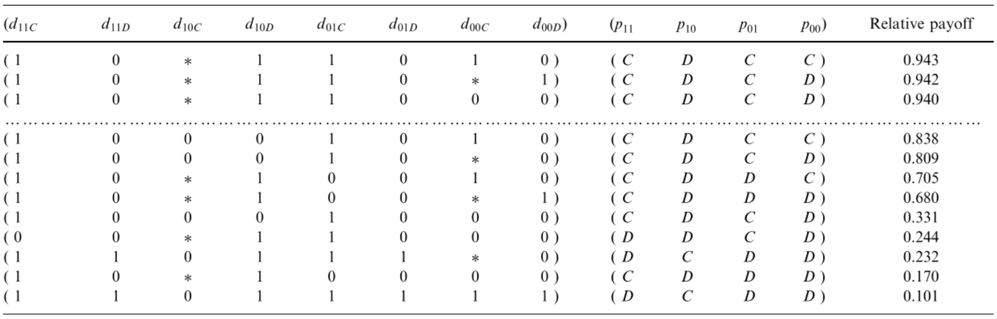
A strategy p is then called evolutionarily stable if it is evolutionarily stable with respect to all strategies $q \neq p$.

Note that we also allow for mixed strategies as well, whereby an agent may choose to play with strategy p, x% of the time and strategy q, $1-x$% of the time. This can be extended to countably finite convex combinations of pure strategies, as defined in [7]. However, so far we have only worked with finite strategy spaces, which leads to finite convex combinations.

Note that the payoff function above is stochastic. This is because opponents / interaction partners are chosen at random and therefore the strategy of the opponent is not deterministic. Note that in our evaluation we assume a uniform distribution for partner choice.

The concept of evolutionarily stable strategies originated in evolutionary biology. Intuitively it means that a population of players with a particular strategy p, if invaded by a minority of players with a new/different strategy q (genetic mutants), is resistant to the propagation of this strategy as a superior one (spread of genetic mutation). Applied to the context of P2P file-sharing, this means that no subset of cheaters can overrule the network, making it unusable for honest/cooperative players, through a dishonest strategy.

Ohtsuki et al. (2004) introduce a direct and an indirect observation model. In our case, because of TrustChain we can assume a direct observation model [17].



(d_{11C}	d_{11D}	d_{10C}	d_{10D}	d_{01C}	d_{01D}	d_{00C}	d_{00D})	(p_{11}	p_{10}	p_{01}	p_{00})	Relative payoff
(1	0	*	1	1	0	1	0)	(*C*	*D*	*C*	*C*)	0.943
(1	0	*	1	1	0	*	1)	(*C*	*D*	*C*	*D*)	0.942
(1	0	*	1	1	0	0	0)	(*C*	*D*	*C*	*D*)	0.940
…	…	…	…	…	…	…	…	…	…	…	…	…
(1	0	0	0	1	0	1	0)	(*C*	*D*	*C*	*C*)	0.838
(1	0	0	0	1	0	*	0)	(*C*	*D*	*C*	*D*)	0.809
(1	0	*	1	0	0	1	0)	(*C*	*D*	*D*	*C*)	0.705
(1	0	*	1	0	0	*	1)	(*C*	*D*	*D*	*D*)	0.680
(1	0	0	0	1	0	0	0)	(*C*	*D*	*C*	*D*)	0.331
(0	0	*	1	1	0	0	0)	(*D*	*D*	*C*	*D*)	0.244
(1	1	0	1	1	1	*	0)	(*D*	*C*	*D*	*D*)	0.232
(1	0	*	1	0	0	0	0)	(*C*	*D*	*D*	*D*)	0.170
(1	1	0	1	1	1	1	1)	(*D*	*C*	*D*	*D*)	0.101

Figure A.3: ESS Pairs (taken from [18])

A.1.3. Leading 8

Ohtsuki (2004) identify a set of ESS strategies, which they refer to as the *leading eight*, which have a relative payoff of over 94%. These are ESS pairs regardless of the cots-to-benefit ratio of transactions, so long as $b > c$. This is not the case for any other pair (d, p). They are also ESS, independently of error rates. The leading 8 are characterised by the following properties, which they all satisfy.

d_{*1C}:	d_{*1D}:	d_{10D}:	d_{11D}:	d_{01D}:
$= 1$	$= 0$	$= 1$	$= 0$	$= 0$

Figure A.4: Leading 8 strategies (taken from [18]).

During our research visit to Sokendai Graduate School of Advanced Studies it was our goal to discuss these leading 8 with the authors of [18] and determine how we could apply these concepts to our research of facilitating cooperation in P2P filesharing networks. The idea was to determine the crucial properties of the leading 8 and determine a reputation mechanisms in P2P filesharing networks that would satisfy these in the hopes that it would facilitate cooperation in that setting as well, i.e. prevent lazy freeriding. The problem we stumbled upon was that the reputation dynamics defined above were all binary, however we wanted our accounting mechanisms to be continuous, such that we could set up a ranking of nodes in the network, for agent to decide whom to contribute to. All attempts to make the upper reputation dynamics continuous without losing the cooperation-facilitating properties were in vein and we decided to pursue research in a different direction. Another problem we incurred was the fact that the reputation dynamics given above were all global. By this we mean that agents held a reputation value that all other agents in the network agreed upon. However, our accounting mechanisms were designed to be personalised, i.e. every agent assigns other agents in the network respective trust scores. It even occurred to us that for global reputation values in a networks it was impossible to achieve any kind of sybil-proofness. We could not reconcile these issues with our topic of research.

A.2. Sybil Resistance Based on Physical Proximity

Recall that while incentivising cooperation through an accounting mechanism was our primary focus, we had to prevent agents from gaming the accounting mechanism through sybil attacks as introduced in chapter 3. We wanted to be able to determine whether a group of nodes is, in fact, controlled by one and the same entity. It's extremely difficult to do this, only by looking at the subjective work graph of a participant. After considering a number of different attributes of nodes in the network that may be helpful in determining whether a set of nodes belong to the same agent in the network, we concluded that one such attribute may be their geographic location. We can determine whether a set of nodes is controlled by the same entity by determining and cross-referencing their respective IP adresses, or location in the Internet layer. In order to keep it as generic as possible, we introduced the notion of a *similarity vector* as a vector of attributes of a set of nodes that may hint at their likelihood of being controlled by the same entity. These could include values such as IP-address, Ping times from different established nodes as well as properties such as traceroutes, etc.

Definition A.2.1 (Similarity Vector)**.**
Given the set of agents in the network V and an agent $i \in V$, we call p_i a similarity vector if it has a set of properties such that if another set of sybil identities created by i, $S_i = \{s_{i1}, \dots, s_{in}\}$ all satisfy $\| p_{s_{ij}} - p_i \| < \varepsilon$ f.a. $j \in \{1, \dots, n\}$ for a given, fixed $\varepsilon > 0$. The exact properties of such a similarity vector will be discussed later.

Definition A.2.2 (Proximity Graph)**.**
Given a set of vertices V and a set of corresponding similarity vectors $\left\{p_i \mid i \in V\right\}$ we derive what we call a proximity graph $G_{Pr} := (V, E)$, which is an undirected, unweighted graph, whereby for $i, j \in V$ we set $(i, j) \in E \Leftrightarrow \| p_i - p_j \| < \varepsilon$. Note that the proximity graph being undirected implies $(i, j) \in E \Leftrightarrow (j, i) \in E$.

Nodes need to be able to cross-reference the similarity vectors (IP-adresses, etc.) of a group of nodes they find suspicious and subsequently group together nodes that are likely to be controlled by a single identity. Analogously to the aforementioned work and trust graphs, agents do not have full knowledge on the entire proxmity graph either. Instead, nodes have a subjective proximity graph, based on agents sharing/reporting their own respective similarity vectors to one another. Agents construct a subjective proximity graph based on the information that is available to them.

Definition A.2.3 (Agent Information)**.**
Every node 'knows' about a subset of all similarity vectors in the network, i.e. agent i has a set $S_i := \left\{p_j^{(i)} \mid j \in V^{(i)}\right\}$ with $V^{(i)} \subset V$.

From this subjective agent information every node constructs its own subjective proximity graph.

Definition A.2.4 (Subjective Proximity Graph)**.** Given a proximity graph $G_{Pr} = (V, E)$ with similarity vectors $\left\{p_j \mid j \in V\right\}$ an agent i with agent information S_i has the subjective proximity graph $G_{Pr}^{(i)} = (V^{(i)}, E^{(i)})$ with $V^{(i)} \subset V$ and $E^{(i)} := \left\{(i, j) \mid \| p_j^{(i)} - p_i \| < \varepsilon\right\}$.

Note that the individual input values of this similarity vector may vary and can be determined based on what is deemed important information. In the case of the Tribler networks one should definitely include IP address, i.e. location in the network graph, as well as ping times from different nodes. One may even want to include some established nodes that are considered trustworthy who will ping new agents and report respective ping times to all other nodes. The point is that every component of the similarity vector will have some notion of a norm on it, i.e. one can measure distance in terms of ping times or in terms of hops in the traceroute tree. This norm will be applied to the similarity vectors to determine the neighbourhoods of nodes in the proximity graph.

Now knowing that we have two respective graphs that we can work with, we introduce a two-layered trust model.

A.2.1. A Two-Layered Trust Model

Our model so far has consisted of an agent i's subjective work graph G_i, a choice set C_i, an accounting mechanism $S^M(G_i, C_i)$ and an allocation policy $A_i(S^M(G_i, C_i))$. From this information, we derived a node or a set of nodes for i to contribute to.

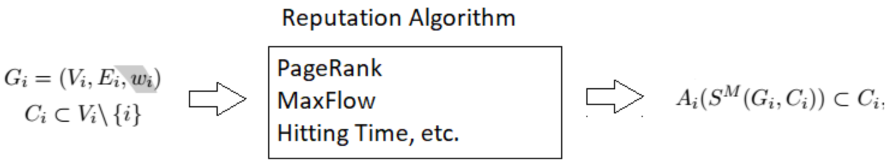


Figure A.5: One-Layer Trust Model

Now, we have a two-layered trust model, in which we are given a work graph and, using our notion of a similarity vector, we determine a proximity graph. Using this proximity graph we derive a newly formed subjective work graph. Then, using our existing accounting mechanisms and allocation policies, we determine a set of nodes to contribute to, analogously to our previous one-layer-model.

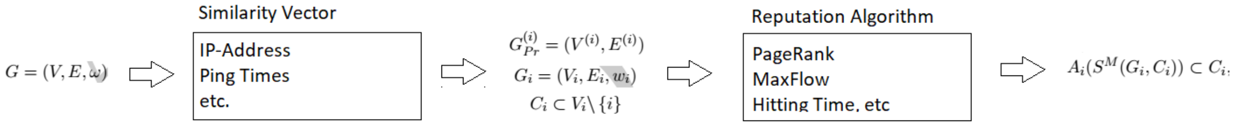


Figure A.6: Two-Layer Trust Model

The idea is that using our proximity graph, we derive our subjective work graph, by collapsing all nodes in the subjective work graph that are connected in the proximity graph.

Definition A.2.5 (Collapsing nodes in the work graph)**.** Given a subjective work graph of agent i, $G_i = (V_i, E_i, w_i)$ and a subjective proximity graph of i, $G^{(i)}_{Pr} = (V^{(i)}, E^{(i)})$, with $V_i = V^{(i)}$, i derives a new subjective work graph from $G^{(i)}_{Pr}$, by collapsing all nodes in G_i that are connected in $G^{(i)}_{Pr}$. This means we obtain a new graph $G'_i := (V'_i, E'_i, w'_i)$ with $|V'_i| \leq |V_i|$ such that

$$\forall i \in V^{(i)}, \neg\exists j \in V^{(i)}, (i,j) \in E^{(i)} : i \in V'_i \;\&\; (i,j) \in E'_i \;\&\; w'_i(i,j) = w_i(i,j).$$

$$\forall i \in V^{(i)}, \exists j \in V^{(i)}, (i,j) \in E^{(i)} : \forall j \in N_{pr}(i) \cup \{i\}, j \notin V'_i, \forall j,k \in N_{pr}(i), (j,k) \in E_i : (j,k) \notin E'_i, \tilde{i} \in V'_i.$$

This means that, any node in the subjective work graph, which does not have any neighbours in the proximity graph is adopted into the new subjective work graph, while any nodes that are connected in the proximitiy graph, are assigned a "proxy" node $\tilde{i}$ in the new work graph, and all edges connecting nodes in a neighbourhood of the subjective work graph are dropped. The outgoing and incoming edges of the neighbourhood of i are now attached to $\tilde{i}$ in the new subjective work graph.

Now we have obtained a new subjective work graph in which all sybil nodes (with a given probability) have been collapsed into a single node. Given this new subjective work graph, agents can run the same accounting mechanisms as before and determine agents' respective accounting values and then determine who to contribute to. If any node that has been removed or collapsed in the subjective work graph now queries an honest node for data, it will not be served and agents that attempt to boos their accounting values through sybil attacks will not be able to increase these values significantly, provided the similarity vectors and the norms have been rigorously defined.

If the similarity vector only consists of IP-addresses we aim to find a mechanism with which nodes can prove to have different IP-addresses from other nodes without revealing their identities. For this we work with the concept of a hash function from the space of possible IP-addresses to the space of public keys.

Definition A.2.6 (Hash Function)**.**
Let $PK := \{\}$ be the set of public keys in the P2P network *tribler* and let $\{0,1\}^{128}$ be the set of IPv6 addresses, comprising 128-bit values. We define the hash function $H : PK \to \{0,1\}^{128}$ as a one-way encryption function satisfying the following 3 conditions:

- Preimage Resistance: Given a value y in the codomain of H, it should be computationally infeasible to determine a value $x \in PK$ such that $H(x) = y$. More precisely, in our case it should take $\mathcal{O}(2^{128})$ time to determine the preimage of y.
- Second Preimage Resistance: Given a value x in the domain it should be equally difficult to determine another x' satisfying $H(x) = H(x')$.
- Collision Resistance: Given our hash function it should be computationally infeasible to determine two values x and x' such that $H(x) = H(x')$. To find such a collision an expected $\sqrt{2^{64}}$ tries are needed. This is due to the birthday paradox, which is introduced by [27].

Definition A.2.7 (Neighbourhood of a node)**.**
A node i that bootstraps in the network then computes the hash of its IP-address and determines the value in the space of possible public keys in the tribler network. Now the node finds all nodes in the network whose public key values are within a given radius δ of $H(x(i))$ whereby $x(i)$ is i's IP-address, i.e. $x : PK \to \{0,1\}^{128}$. Then we obtain

$$N(i) := \left\{ j \in V_i \mid \| j - H(x(i)) \| \le \delta \right\}.$$

This is what we call the neighbourhood of node i. Note, however that this is not the same as a neighbourhood of a node in the interaction graph. Instead, we introduce a new graph, namely the *Hash Graph.*

Definition A.2.8 (Hash Graph)**.**
Given a work graph $G = (V, E, w)$ with nodes $V \subset PK$, we derive an undirected and unweighted graph from the neighbourhoods determined above. We obtain

$$E = \left\{ (i, j) \in V \times V \mid j \in N(i) \right\}.$$

Note that it holds $i \in N(j) \Leftrightarrow j \in N(i)$.

The idea behind this is that if an agent i decides to create a set of fake identities $\{s_{i1}, \dots, s_{in}\}$ then these identities will all have the same IP-address and therefore all will have the same hash

$$h(x(i)) = h(x(s_{ij})) \text{ f.a. } j \in \{1, \dots, n\}.$$

This will lead to a very big neighbourhood in the hash graph, which will be noticeable and collapsing all of these nodes will render the sybil attack unbeneficial.

At this point, we felt that we had deviated too far from the topic a thesis in applied mathematics should have and decided to no longer pursue the line of reasoning. This does not mean that such a strategy could not be effective in mitigating the effects of sybil attacks.

Bibliography

[1] Altman, Alon, and Moshe Tennenholtz. "Ranking systems: the PageRank axioms." Proceedings of the 6th ACM conference on Electronic commerce. ACM, 2005.

[2] Botsman, Rachel. "Big data meets Big Brother as China moves to rate its citizens." Wired UK 21 (2017).

[3] Bravetti, Alessandro, and Pablo Padilla. "An optimal strategy to solve the Prisoner's Dilemma." Scientific reports 8.1 (2018): 1948.

[4] Brouwer, Jetse. "Consensus-less Security: A truly scalable distributed ledger." Master Thesis. TU Delft 2020. TU Delft Education Repository. Web. Accessed 18.02.2020.

[5] Buechler, Matthew, et al. Decentralized reputation system for transaction networks. Technical report, University of Pennsylvania, 2015.

[6] Cohen, Bram. "Incentives build robustness in BitTorrent." Workshop on Economics of Peer-to-Peer systems. Vol. 6. 2003.

[7] Ferguson, T.S. "Game Theory, Second Edition" (Mathematics Department UCLA, 2014)

[8] Harms, Jan-Gerrit. "Creating trust through verification of interaction records." (2018).

[9] Levin, Dave, et al. "Bittorrent is an auction: analyzing and improving bittorrent's incentives." ACM SIGCOMM Computer Communication Review 38.4 (2008): 243-254.

[10] Li, James. "A Survey of Peer-to-Peer Network Security Issues." Retrieved November 29 (2007): 2010.

[11] Liu, Brandon K., David C. Parkes, and Sven Seuken. "Personalized hitting time for informative trust mechanisms despite sybils." Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems. International Foundation for Autonomous Agents and Multiagent Systems, 2016.

[12] Meulpolder, Michel, et al. "Bartercast: A practical approach to prevent lazy freeriding in p2p networks." 2009 IEEE International Symposium on Parallel & Distributed Processing. IEEE, 2009.

[13] Nakamoto, Satoshi. Bitcoin: A peer-to-peer electronic cash system. Manubot, 2019.

[14] Newman, Mark EJ. "Detecting community structure in networks." The European Physical Journal B 38.2 (2004): 321-330.

[15] Nowak, Martin A. "Five rules for the evolution of cooperation." science 314.5805 (2006): 1560-1563.

[16] Ohtsuki, Hisashi, et al. "A simple rule for the evolution of cooperation on graphs and social networks." Nature 441.7092 (2006): 502.

[17] Ohtsuki, Hisashi, and Yoh Iwasa. "How should we define goodness?—reputation dynamics in indirect reciprocity." Journal of theoretical biology 231.1 (2004): 107-120.

[18] Ohtsuki, Hisashi, and Yoh Iwasa. "The leading eight: social norms that can maintain cooperation by indirect reciprocity." Journal of theoretical biology 239.4 (2006): 435-444.

[19] Otte, P. "Sybil-resistant trust mechanisms in distributed systems." (2016).

[20] Otte, Pim, Martijn de Vos, and Johan Pouwelse. "TrustChain: A Sybil-resistant scalable blockchain." Future Generation Computer Systems (2017).

[21] Pouwelse, Johan A., et al. "TRIBLER: a social-based peer-to-peer system." Concurrency and computation: Practice and experience 20.2 (2008): 127-138.

[22] Page, Lawrence, et al. The PageRank citation ranking: Bringing order to the web. Stanford InfoLab, 1999.

[23] Seuken, Sven, and David C. Parkes. "On the Sybil-proofness of accounting mechanisms." (2011).

[24] Seuken, Sven, Jie Tang, and David C. Parkes. "Accounting mechanisms for distributed work systems." Twenty-Fourth AAAI Conference on Artificial Intelligence. 2010.

[25] Seuken, Sven, et al. "Work accounting mechanisms: Theory and practice." Working Paper. Department of Informatics. University of Zurich, 2014.

[26] Seuken, Sven, and David C. Parkes. "Sybil-proof accounting mechanisms with transitive trust." Proceedings of the International Foundation for Autonomous Agents and Multiagent Systems (2014).

[27] Smart, Nigel P. Cryptography made simple. Vol. 481. Cham: Springer, 2016.

[28] Stannat, Alexander, and Johan Pouwelse. "A Random Walk based Trust Ranking in Distributed Systems." arXiv preprint arXiv:1903.05900 (2019).

[29] Tang, Jie, Sven Seuken, and David C. Parkes. "Hybrid transitive trust mechanisms." Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems: volume 1-Volume 1. International Foundation for Autonomous Agents and Multiagent Systems, 2010.

[30] Tanenbaum, Andrew S., and Maarten Van Steen. Distributed systems: principles and paradigms. Prentice-Hall, 2007.

[31] van den Heuvel, Bram, and Dai, Yinghao. "Trust in Distributed Systems." (2019).

[32] andy@torrentfreak.com "The Early Days of Mass Internet Piracy Were Awesome Yet Awful". `https://torrentfreak.com/the-early-days-of-mass-internet-piracy-were-awesome-yet-awful-180211/` (11.02.2018)

[33] PsychologyProf@gmail.com "Game Theory – best strategy for Prisoner's Dilemma used in advice column for real life situation" `http://cogsciandtheworld.blogspot.com/2009/10/game-theory-best-strategy-for-prisoners.html` (11.10.2009)