THE UNIVERSITY
of EDINBURGH

Incentives in Blockchain Protocols

Aikaterini-Panagiota Stouka

Doctor of Philosophy

Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh

2021

Abstract

Bitcoin is a digital cryptocurrency supported by the blockchain protocol proposed by Nakamoto in 2008. The blockchain protocol offers a public transaction ledger, organized as a sequence of blocks of transactions. The sequence of blocks is maintained in a distributed way by a set of peers called miners. The main novelty of the Nakamoto's protocol is the use of a proof of work scheme in which miners expend computational power to get a chance to produce a new block and in turn they get a reward for each block they produce. The success of Bitcoin prompted a large variety of other blockchain protocols that attempt to improve on various aspects of the original protocol. Two examples that are related to the present thesis are (i) variants of the original proof of work, longest chain protocol that attempt to improve its resilience characteristics with respect to adversarial behavior and (ii) proof of stake blockchain protocols according to which each participant is elected to produce a block with probability proportional to its stake, rather than computational power. In this thesis we examine blockchain protocols from a game theoretic perspective. This means that we consider participants as rational utility maximizers as opposed to being divided between honestly behaving and adversarial. This thesis mainly focuses on answering the following three questions: (i) do miners have incentives to follow the blockchain protocol, when all the other participants do so (this is related to the notion of Nash equilibrium) (ii) how can we design a reward mechanism that promotes decentralisation, by disincentivising the formation of undesirable large pools in proof of stake blockchain protocols? (iii) given such a reward mechanism, how can we disincentivise existing pools to create a cartel and censor other pools' registration in the blockchain with the aim to avoid competition? In order to answer the first question we propose a suitable notion of Nash equilibrium, called "coalition-safe equilibrium with virtual payoffs (EVP)". This notion allows us to provide (i) a unified picture of the incentives in the Bitcoin blockchain protocol when the participants are rational and try to maximize various utilities based on the rewards and the costs, and (ii) novel results regarding incentives in a fair variant of the Bitcoin protocol called Fruitchain [PODC 2017, Rafael Pass et al.]. The motivation for the second question that this thesis answers is the following: although Bitcoin was designed to be executed in a decentralised way without a trusted party, participants tend to avoid participating directly in the protocol. Instead, they tend to create teams,

called pools, which are managed usually by a single participant, called pool leader and they follow pool leader's instructions in order to get paid. For example, very few pools may have collectively the majority of computational power, something that could be dangerous for the security of Bitcoin if the operators of these pools collude. In order to answer the second question we examine how participants in a proof of stake blockchain protocol should be rewarded so that in a Nash equilibrium they form k pools where k is a parameter. To be more specific, we define what a reward sharing scheme (RSS) is and we propose an RSS that achieves the following level of decentralization: (1) it incentivizes participants to form k pools and (2) it mitigates Sybil behavior [IPTPS 2002] that in our case is related to how many independent entities are the actual pool leaders of these k pools. In addition, we provide a formal analysis regarding the equilibria that arise from a system using this RSS. We discuss at some length also the deployment of such an RSS in a proof of stake system. We remark that the reward mechanism that was implemented in the incentivised testnet and the "Shelley update"' launched by the company IOHK (Input Output) on the Cardano cryptocurrency was based on our results. The third question we answer thoroughly and formally in this thesis relates to a serious concern that arises in the deployment of an RSS and relates to censorship of transactions. In a proof of stake system in order for a pool to be registered it should create a special transaction and this transaction should become part of the ledger in order to be actionable. However, the existing pools that run the blockchain protocol may not be willing to add such a transaction, i.e., engage in censorship. We provide an anti-censorship mechanism and we prove the favorable equilibria that arise when such a mechanism is utilized.

Acknowledgements

I would like to express my gratitude to my supervisor Prof. Aggelos Kiayias for entrusting me and giving me the opportunity to become a member of the Blockchain Technology Laboratory (BTL) and the CryptoSec group. I am grateful for his guidance, his support and our fruitful discussions during my PhD studies that evolved me and shaped me as researcher.

I would like to express my gratitude to Prof. Elias Koutsoupias, with whom we collaborated closely during my PhD studies, because he besides my supervisor introduced me to the area of incentives, an area that really inspired me as a researcher. I thank him for his support and his guidance.

I would like also to thank Dr. Lars Bruenjes, Education Director of the IOHK (Input Output) technology company, for our collaboration. The collaboration with him helped me to evolve as researcher and understand how well the theory with the experiments can be combined.

I would like to thank all the IOHK members and especially Dr. Philipp Kant and Dr. Duncan Coutts with whom we had fruitful discussions regarding the deployment of our reward scheme in the Shelley update launched by the company IOHK.

I would like to thank my second supervisor Prof. Myrto Arapinis for her valuable comments and suggestions as a member of my annual review process.

I would like to thank professors Aris Pagourtzis, Stathis Zachos,Alexandros Arvanitakis,Lefteris Kirousis and in general all the professors who inspired me during my undergraduate and post graduate studies.

I would like to thank all the members of the Blockchain Technology Laboratory, the CryptoSec group and the Laboratory for Foundations of Computer Science (LFCS) that made my working environment so pleasant.

I would like to thank Dr. Mirjam Wester, senior research manager of the IOHK company and Gareth Beedham, Senior Administrative Secretary of BTL for their administration support.

Moreover I would like to thank my internal examiner Prof. Kousha Etessami and my external examiner Prof. Georgios Christodoulou for their valuable comments.

Finally I would like to thank my family and friends for their endless support.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

Some chapters of the thesis are based on the following papers:

- Lars Brünjes, Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. Reward sharing schemes for stake pools. In 2020 IEEE European Symposium on Security and Privacy, pages $256 - 275$, Los Alamitos, CA, USA, sep 2020. IEEE Computer Society. Full version available at arXiv, CoRR, abs/1807.11218, 2018.

- Aggelos Kiayias and Aikaterini-Panagiota Stouka. Coalition-safe equilibria with virtual payoffs. arXiv, CoRR, abs/2001.00047. Under submission. 2020.

- Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. Incentives Against Power Grabs or How to Engineer the Revolution in a Pooled Proof of Stake System, Under submission. 2020.

- Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. Reward Sharing Schemes for Collaborative Projects. Manuscript. 2020.

(*Aikaterini-Panagiota Stouka*)

To the memory of my father

Table of Contents

List of Figures

Chapter 1

Lay Summary

Bitcoin is a digital currency proposed by Nakamoto that uses a blockchain protocol to form a distributed transaction ledger stored and maintained by peer to peer nodes. This ledger consists of blocks that include transactions between users. Each block refers to the previous block creating a blockchain. The transactions record who sent money (Bitcoin) to whom and they are ordered in the ledger. Thus the ledger determines how many Bitcoins each user owns. The ledger is extended by one block when a node, called miner, solves a cryptographic puzzle spending computational power and performing proof of work. The miner who extends the ledger earns an amount of Bitcoins minted when the block is added to the ledger. Note that the Bitcoin ledger is stored and extended by peer to peer nodes and not by a trusted party (a single party that we should trust).

After the success of Bitcoin, a large variety of other blockchain protocols were proposed that attempt to improve on various aspects of the original protocol. Two examples that are related to the present thesis are (i) the Fruitchain blockchain protocol that forms a fair chain where the fraction of blocks produced by each miner is very close to its relative computational power and (ii) proof of stake blockchain protocols according to which the ledger is extended without the need of computational power.

In this thesis we examine the incentives in blockchain protocols or in other words we study the blockchain protocols from a game theoretic perspective. This means that: (i) we consider the participants/nodes as rational in the sense that they behave in a way that maximizes a quantity that they find important, called utility, for example their profit. (ii) we examine if the participants have incentives to follow the protocol. A notion related to this is Nash equilibrium. A joint

strategy (a vector that describes the behaviour and the choices –strategies- of all the participants) is a Nash equilibrium when the following holds: if all but one participants behave as indicated by the joint strategy, then the remaining participant will not have higher utility if it chooses a strategy different from what is described in the joint strategy. Note that if the participants suddenly chose a joint strategy that is Nash equilibrium, then nobody would have incentives to change its strategy.

So when we say that we examine the incentives in a protocol, we mean that we search for joint strategies that are Nash equilibria, and we evaluate what characteristics the system will have if the participants choose this joint strategy.

Towards this direction we propose a framework, based on the Nash equilibrium notion, which is suitable for blockchain protocols, and we use it to examine the incentives in the Bitcoin and the Fruitchain blockchain protocol.

In Bitcoin there is the tendency that miners form huge pools where each pool is controlled by a miner. This endangers the decentralization of the system, because we should trust the miners who control the pools that they will behave as intended. In this thesis we prove that if we use the reward mechanism of Bitcoin in proof of stake blockchain protocols, then there is no equilibrium with more than one pool. Moreover, we propose a reward mechanism that disincentivizes the formation of huge pools in proof of stake blockchain protocols ensuring a level of decentralization (these results are based on our work published in IEEE European Symposium on Security and Privacy 2020 conference). The reward mechanism that was implemented in the Shelley update on the Cardano proof-of-stake blockchain platform launched by the company IOHK (Input Output) technology company was based on our reward mechanism.

In many proof of stake blockchain protocols, in order for a pool to be created, a special transaction should be added to the ledger. The ledger is extended by the participants who control the pools, called pool leaders, and run the blockchain protocol. Thus pool leaders may have incentives to create a cartel and ignore transactions that register new pools with the aim to avoid competition. We propose an anti-censorship mechanism that incentivizes at least one pool leader to include these types of transactions.

Chapter 2

Introduction

Bitcoin is a digital currency that was proposed by Nakamoto in 2008 [100] and was formally analyzed in the cryptographic setting in [59, 107]. It uses a blockchain protocol that formulates a distributed ledger. This ledger consists of transactions that are organised into blocks and is distributed among peer-to-peer nodes. The blocks are connected in the sense that each block refers to the previous block and form a chain.

Each block is produced when a node solves a *cryptographic puzzle* [14, 45, 73, 111] via a procedure that is called *proof of work*. This procedure requires computational power. The node who solves a proof of work puzzle and produces a block that extends the ledger is rewarded with an amount of Bitcoins minted the time the block is produced. In case more than one chains have been created, the longest chain constitutes the ledger. In a nutshell, *proof of work* guarantees that with very high probability a node who deviates from the Bitcoin protocol cannot produce an alternative chain (a chain that differs from the ledger even if we prune the last blocks of the chain and the ledger) that is longer than the existing ledger without using massive amount of computational power.

Under certain assumptions, in a setting where there exist (i) honest participants who follow the protocol and (ii) malicious participants (fewer than the honest) who can deviate from the protocol arbitrarily, the Bitcoin protocol satisfies the following two properties: *common prefix property* and *chain quality property* [59]. In high level, the *common prefix property* guarantees that with very high probability the chains that the honest participants consider as ledger (or in other words their *local chains*) have in common their initial part. This means that with very high probability an honest participant will not adopt as

ledger a chain that differs from its current local chain in its initial part. Thus it holds that with very high probability the malicious participants cannot produce an alternative chain that differs from the current honest participants' local chains in their initial part and is longer than them. The *chain quality property* provides a lower bound on the blocks that are produced by honest participants. Note that the common prefix property does not provide any guarantee for the last part of each participant's local chain. Thus the "view" of the honest participants on the recent blocks can differ (for more details regarding Bitcoin and the properties it satisfies cf. Chapter 3).

Although these two properties can give us a clear picture for the Bitcoin protocol from a security point of view considering honest and malicious participants, they do not capture rational behaviors where the participants are neither honest nor malicious; they are *rational* in the sense that they act in a way that maximizes their profit or another quantity that they find important called *utility*.

When we study a blockchain protocol considering the participants as rational, then we say that we study the *incentives* in a blockchain protocol.

Despite the fact that Bitcoin satisfies the common prefix and chain quality property, it has some vulnerabilities that other blockchain protocols proposed later tried to overcome. Some examples related to the present thesis are (i) the Fruitchain blockchain protocol [108] that formulates a chain that is *fair* in the sense that the fraction of the blocks in the chain belonging to each coalition of miners is guaranteed to be near its relative computational power and (ii) *proof of stake* blockchain protocols (for example [19, 30, 42, 80, 98]) according to which the ledger is extended without the need of computational power. In high level in a proof of stake blockchain protocol the participants get elected to produce blocks with probability that is related to their relative *stake* where relative stake is the relative amount of currency they own.

2.1 Contribution

In this thesis we study the incentives in blockchain protocols. In more detail, this thesis focuses on answering the following three questions: (i) will rational miners follow the Bitcoin and the Fruitchain blockchain protocol [108] if all the other participants do so (this is related to the notion of *Nash equilibrium*) (Chapter 4) (ii) how can we design a reward mechanism that disincentivizes the formation

of huge *pools* in *proof of stake* cryptocurrencies protecting their decentralised nature? (Chapter 5) and (iii) if we have such a reward mechanism, how can we disincentivize existing pools to create cartels and censor the registration of new pools in the blockchain with the aim to avoid competition (Chapter 6)?

In order to answer the first question when the participants try to maximize different type of utilities, we propose a suitable notion of *Nash equilibrium*, called *coalition-safe equilibrium with virtual payoffs (EVP)*[1]. The use of a common framework for various utilities is important because a few previous works had used different models to study incentives in Bitcoin and some of them ended up in seemingly contradictory results regarding whether Bitcoin is *incentive compatible* or not (cf. Chapter 4 for more details).

Moreover, our framework, compared to previous works, at the same time (i) considers the fact that participants may have a divergent view on the rewards of the other participants (thus we call the rewards *(virtual)*) (ii) is well defined without assuming that the blockchain protocol we examine satisfies specific security properties such as the common prefix property (iii) can be used for different types of utility based on (a) the rewards the participants earn for their participation in the protocol and (b) the cost they incur (iv) ensures the following: even when a coalition can increase its utility with small (but not negligible) probability by deviating then the protocol does not satisfy our notion.

This framework allows us to provide a unified picture of the incentives in the Bitcoin protocol considering various utilities and novel results regarding incentives in the Fruitchain protocol [108].

In more detail, regarding incentives in [108], we prove that our notion of equilibrium is satisfied by the Fruitchain blockchain protocol for coalitions up to $n-1$ where n is the number of all the participants when the utility is based on *absolute rewards minus absolute cost* and for coalitions smaller than $n/2$ when the utility is based on *relative rewards*. With the term absolute rewards we refer to the amount of the rewards that a set of participants receives. With the term absolute cost we mean the cost that this set of participants pays expressed in absolute terms. With the term relative rewards we refer to the rewards of this set of participants divided by the total rewards given to all the participants. In addition we define a property called *weak fairness*, which is weaker than *fairness* proposed in [108] and we prove that a protocol that satisfies this property satisfies

[1] These results are based on our paper [81].

also our notion for coalitions smaller than $n/2$ when the utility is based on relative rewards.

The motivation for the second question this thesis answers is the following: although Bitcoin was designed to be carried out in a decentralised way without a trusted party, the miners tend to avoid participating directly in the protocol. Instead, they tend to create teams, called *pools*, which are managed usually by a single participant, called *pool leader* and they follow pool leader's instructions in order to get paid. For example, according to [4] very few pools have the majority of the computational power something that could be dangerous for the security of Bitcoin if the pool leaders of these pools colluded.

In order to answer the second question we examine how participants in a *proof of stake* blockchain protocol should be rewarded so that in a Nash equilibrium they form k pools where k is a parameter[2]. To be more specific, we define what a reward sharing scheme (RSS) is and we propose an RSS that achieves the following level of decentralization:

1. it incentivizes participants to form k pools of equal size. This means that the creation of k pools of equal size is a Nash equilibrium.

2. it mitigates *Sybil behavior* [43] that in our case is related to how many entities are the actual pool leaders of these k pools. Observe that this is important because if we had k pools but most of them were controlled by the same physical identity then our system would not be decentralized and also its security could be hurt. In more detail, in our case Sybil behavior refers to a user who opens/creates multiple pools using different identities that attract the majority of the stake. Such a user could invalidate the security properties of the underlying proof of stake protocol that hold under the assumption that the participants who can deviate from the protocol control less than half of the total stake.

3. it offers a trade off between protection against Sybil attacks that are related to decentralization and cost efficiency. In more detail, in the extreme case where we do not care about Sybil protection, the pool leaders in a Nash equilibrium will be the participants with the lowest cost. Otherwise, the stake of the pool leader affects the rewards of each pool and thus the pool

[2]These results are based on our paper [29] published in the IEEE European Symposium on Security and Privacy 2020 and the extended version of this paper in [28].

leaders in a Nash equilibrium are the participants with the most profitable (for their pools) combination of cost and stake.

An RSS consists of two levels. The first level includes a reward function that determines how many rewards a pool will get at the end of a time interval called *epoch* and the second level determines how these rewards will be shared inside the pool.

Before we describe our RSS, we justify why simple reward functions cannot provide us with the guarantees described above. In more detail, we prove that if we use a specific class of reward functions, then there are no equilibria with more than one pools.

This class of reward functions includes the *fair* reward function where the pools take a fraction of the available rewards that is proportional to their relative computational power in a proof of work protocol or their stake in a proof of stake protocol. This reward function describes how the participants in Bitcoin get rewarded in case all participants are honest (they always follow the protocol) and how the participants in the Fruitchain blockchain protocol [108] get rewarded (if we do not take into account the stochastic nature and the variance of the rewards).

Moreover, we prove that for specific classes of reward functions there are instances for which we have the other extreme case where there are no Nash equilibria with a number of pools smaller than the number of participants.

After describing our RSS, we provide a formal analysis with the Nash equilibria that arise from a system using this mechanism. Finally we discuss the deployment of such an RSS in a proof of stake cryptocurrency. Furthermore

1. one of the co authors of our paper [29] Lars Bruenjes ran experiments which demonstrate that a system using our RSS converges fast to the Nash equilibria we describe considering a setting where the participants react to each other's strategic moves over an indefinite period of interactive play.

2. the reward mechanism that was implemented in the incentivised testnet and the Shelley update on the Cardano mainnet launched by the company IOHK [3] (Input Output) was based on these results.

The third question we answer in this thesis is an aspect of the deployment

of our RSS in a proof of stake protocol, called *censorship*[3]. In a proof of stake cryptocurrency in order for a pool to be registered, it should create a special transaction and this transaction should become part of the ledger. However, the existing pools who run the blockchain protocol may not be willing to add such a transaction when they are elected to produce a block because the want to avoid competition. In this case they create a cartel. To solve this we propose an anti-censorship mechanism that

1. uses some cryptographic functions of the underlying blockchain protocol.
2. uses the fact that information is diffused in a peer-to-peer network.

Finally we describe and prove which are the Nash equilibria that arise when we use such a mechanism.

[3]These results are based on our under submission paper "Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. Incentives Against Power Grabs or How to Engineer the Revolution in a Pooled Proof of Stake System, 2020".

Chapter 3

Background

3.1 Blockchain Protocols

3.1.1 Bitcoin

Bitcoin [100] is a digital currency that uses a blockchain protocol in order to formulate a distributed ledger. This ledger has the following characteristics:

1. it consists of blocks that include transactions between users. Each block refers to the previous block creating a blockchain and is produced by nodes called *miners* who spend computational power performing *proof of work* in order to solve a cryptographic puzzle [14, 45, 73, 111].

   The transactions record who sent money (Bitcoins) to whom and they are ordered in the ledger. Thus the ledger determines how many Bitcoins each user owns. Two relevant questions are the following:

   - how can a user earn Bitcoins apart from receiving it by another user? Or in other words how are Bitcoins created?

     The answer is that when a miner creates/produces a block that is added to the ledger, then it gets rewarded with a flat amount of Bitcoins that is minted the time the block is produced. So the first Bitcoins were created when the first blocks were produced. Observe that this reward motivates the miners to spend computational power in order to produce blocks and extend the ledger. Adding to the flat reward, the miner who extends the ledger gets rewarded with transaction fees which is the "tip" that the user who issues the transaction

gives to the miner.

- is the information about how many Bitcoins each users owns public?

  The answer is that Bitcoin is pseudonymous in the sense that users do not use their real identity but a public key when they receive Bitcoins by another user. Furthermore they are expected to use a different public key each time they issue a transaction so that it is not easy for somebody to find the whole amount of Bitcoins they own. [1]

2. due to *proof of work* it is immutable under certain assumptions that we present later in this chapter.

3. it is stored by peer-to-peer nodes. This means that there is no trusted party that stores and extends this ledger.

The following background information is based on [1, 100, 101].

3.1.1.1 Fork

If two or more miners have produced a block of the same level (when we say that two blocks belong to the same level we mean that they both extend the same block) then only one of these blocks can be included in the ledger. In such a scenario we say that we have a fork. This means that we have more than one chains that have common an initial part.

When a node receives more than one chain then it considers the longer one as the ledger. If both have the same length then it chooses the first chain it receives (it could chose also at random). Note that each node may have a different local view and consider a different chain (called local chain) as ledger because it may have received different blocks compared to other nodes at a particular time. Under certain assumptions the authors of [59] have proved that all the honest participants share the same initial part of the chain with very high probability, (see *common prefix property* below).

3.1.1.2 Rewards

Summarizing the rewards a miner earns are

[1] There are some attacks that connect public keys that belong to the same owner based on specific patterns. Some works that analyze the statistical properties of the Bitcoin transaction graph and describe identification attacks in Bitcoin are [97, 113].

- the flat reward of each block in the ledger that it has produced. The flat reward of the first block (called *genesis*) was 50 Bitcoins (BTC) and it has been halved every after 210000 blocks. This means that when the first 210000 blocks were created, the reward per block became 25, when the first 420000 blocks were created, the reward became 12.5 and so on.

- the transaction fees that we described earlier.

3.1.1.3 Proof of Work (POW)

In order for a miner to produce a block, it needs to spend computational power to solve a cryptographic puzzle [14, 45, 73, 111] that is related to a hash function (see below for more details). In a nutshell, *proof of work* guarantees that with very high probability a coalition of malicious participants cannot create a very "deep" fork (fork in the initial part of the ledger) without using a great amount of computational power.

Hash Function: it is a function that (i) has as input a string of arbitrary size and as output a string of fixed size (the size is denoted by κ) (ii) it is easy to compute and (iii) difficult to invert. More formally a hash function $H : \{0,1\}^* \to \{0,1\}^\kappa$ satisfies the following two properties:

1. second preimage resistance: for every x it is difficult to find $y \neq x$ such that $H(x) = H(y)$.

2. collision resistance: it is difficult to find x, y such that $x \neq y$ and $H(x) = H(y)$.

The miners who want to extend the ledger compute the hash value of a vector that includes mainly the following:

1. the hash of the last block.

2. a root node of a *Merkle tree* of the existing *valid* transactions.

   Merkle tree is a tree that consists of leafs that include data blocks (in our case these are transactions) and non-leaf nodes that are labelled with the hash value of their child nodes' labels.

   When we say *valid* we mean transactions (i) where the sender owns the Bitcoins that it wants to transfer and (ii) which are structured according

to the protocol. A block is considered valid if it contains valid transactions and also is structured as supposed to be by the protocol.

3. a nonce. This is the part of the input that the miner changes each time it computes a hash value in order to produce a block.

Target and Difficulty: when a miner finds a hash value that is smaller than a quantity called *target* then it has found/ produced/created a block. Otherwise it changes the *nonce* and tries again until it finds a hash value that is smaller than the target.

Recall that the longest chain constitutes the ledger, so when a miner creates a block $\boldsymbol{B}$, then all the other honest miners adopt this block in their chain and they change the input that they use for computing hash values (because the last block of the ledger is now $\boldsymbol{B}$ and also new transactions should be included in the Merkle tree).

Note that the smaller the target is, the more difficult is for the miner to find a block. The difficulty of a block refers to the target that is used in the proof of work procedure.

The target for mining blocks adjusts every 2016 blocks so that one block is produced every ten minutes in expectation.

For example, when the total computational power increases then the target decreases (and the difficulty increases) so that blocks continue to be produced every ten minutes in expectation, not more frequently. Note that if the blocks are produced much faster than ten minutes, then there is not enough time for the new blocks to be delivered to the miners. This means that the miners that have not received the new block spend computational power to create blocks of the same level instead of extending the ledger. In a similar way when the total computational power decreases the target increases (and the difficulty decreases) so that the frequency of producing blocks does not get reduced. This is important so that the ledger continues to extend and records transactions in a regular rate.

Note that this algorithm of the target adjustment takes into account the computational power that is used for producing only blocks that have become part of the ledger not blocks that have been kept secret or are in a fork.

At this point we describe in high level the security properties of Bitcoin presented in [59]. We also describe some features of the model used in [59] that we use in this thesis.

3.1.1.4 Security properties

In [59] the core of the Bitcoin protocol is extracted and is used to prove that Bitcoin satisfies *common prefix* and *chain quality* properties and thus it formulates a transaction ledger satisfying *persistence* and *liveness* properties under the following assumptions and model specifications:

1. there are two types of participants: (i) the honest participants who always follow the protocol and (ii) the malicious participants who deviate arbitrarily. To be specific there is an adversary who corrupts in the beginning a number of participants making them malicious.

2. there is a fixed number of participants with equal computational power.

3. the honest participants are more than the malicious (honest majority).

4. the target of blocks is fixed (for variable difficulty see [60]).

5. the protocol is progressing in regular time intervals called *rounds* and is synchronous in the sense that (i) at the end of each round each honest participant receives all the messages sent by all the other honest participants during the round (ii) the total expected number of blocks produced by all the participants during a round tends to zero (much smaller than 0.1). Intuitively, this means that with very high probability during each round at most one block is produced, and thus it is unlikely that more than one blocks have been produced and the participants have not received them (the honest participants get synchronised at the end of each round).

6. The hash function is modelled as random oracle [18].

   Random oracle functionality: when a participant queries this oracle with input x then:

   (a) if it has not asked before the oracle with the same input, the oracle returns a random value y in $\{0,1\}^{\kappa}$ where κ the security parameter and stores (x,y) in a table.

   (b) If it has asked the oracle with the same input then it returns the same output as before.

In the execution model described in [59] each participant can ask this oracle at most q queries per round and the adversary at most q queries for each participant it has corrupted. These queries reflect the computational power of the participants. Verification queries to verify which is the hash of a value are allowed only to the honest participants not to the adversary.

7. The communication between the participants is determined by the following functionality.

   Diffuse Functionality: each participant maintains a RECEIVE tape and at any moment can fetch the contents of this tape. In each round each honest participant determines a message m that it wants to diffuse. This reflects the block it has produced and it can be also an empty string. When each honest participant does so, the functionality asks the adversary if and to whom honest participants it allows the message to be sent (be written in their RECEIVE string). At the end the adversary determines its own messages and the participants to whom it wants these messages to be delivered. At the end of the round the diffuse functionality delivers all the messages that the adversary did not allow during the round.

Note that we do not give more details regarding the execution model that is used in [59] because we describe it in the following chapter.

Before we present the security properties of the Bitcoin protocol (as they have been proved in[59]) we give some definitions that we will need based on [76].

Definition 1 (Negligible function)**.** *A function $f(x) : \mathbb{N} \to \mathbb{R}^+$ is negligible in x if for every positive polynomial $g(x)$ there is $n_0 \in \mathbb{N}$ such that for every x integer such that $x > n_0$ it holds $f(x) < \frac{1}{g(x)}$.*

An example of a negligible function is $1/2^x$.

Definition 2 (Overwhelming function)**.** *A function $f(x) : \mathbb{N} \to \mathbb{R}^+$ is overwhelming in x when $1 - f(x)$ is negligible in x.*

When we say that an event happens with overwhelming probability in the security parameter, we mean that the probability with which this event does not happen is negligible or in other words drops exponentially as the security parameter increases.

Common prefix property: when the number of the malicious participants is enough smaller than the number of the honest participants, then the initial part of the chain that belongs to an honest participant is common in all the other honest participants with *overwhelming probability* in the security parameter of the system.

Note that the better synchronicity the network has, the weaker the assumption regarding the ratio of malicious participants can be. To be specific, when the total expected number of blocks produced per round tends to become zero, then the number of malicious participants can become a little lower than the number of the honest participants, known as *honest majority assumption*.

Chain quality with parameters $l \in \mathbb{N}$ and $\mu \in \mathbb{R}$: in any honest chain (chain that an honest participant considers as ledger) with at least l blocks the ratio of the blocks produced by the honest participants is at least μ. This means that the fraction of the blocks produced by the malicious participants is upper bounded by $1-\mu$. In Bitcoin this upper bound is $t/(n-t)$ where t the number of the malicious participants and n the total number of participants.

You can observe that:

1. when t is smaller than $n/2$ any honest chain will still contain some honest blocks. Note that it is important for the ledger to include honest blocks (blocks produced by honest participants) because otherwise the malicious participants can censor transactions issued by the honest by not including them in the blocks they mine.

2. the lower t is, the more honest blocks are guaranteed to be included in the ledger.

3. this bound is tight when the following happens: in a case where two blocks of the same level have been produced, then the honest participants always choose to extend the adversarial block (block produced by the adversary), if there is one. Observe that according to the Diffuse Functionality the adversary has the option to deliver always its block first. Thus if the honest participants choose to extend always the first block they receive, then they always prefer the adversarial block except the times they have mined their own block. This attack is presented in [59] and is based on an attack called *selfish mining attack* presented in [48].

The attack that makes the bound in the chain quality tight [48, 59]: in high level the adversary tries to extend the ledger, but it keeps the block it produces private until an honest participant produces a block. When an honest participant produces a block, the adversary diffuses one of its blocks. Given that it can deliver its block first, the other honest participants choose to extend the adversarial and not the honest block. If the local chain of the adversary becomes smaller than the local chains of the honest participants, then it adopts the local chain of an honest participant.

This attack gives the adversary the opportunity to:

1. drop honest blocks.

2. reduce the total number of blocks in the ledger and in this way obtain a ratio of blocks in the local chains of the honest participants that is much higher than its *fair* [108] share t/n [48].

3. make the honest participants spend computational power without gaining any profit.

Assuming honest majority and fast synchronization (the total number of blocks produced per round tends to zero) then a blockchain protocol that satisfies the above properties formulates a transaction ledger with the following two properties.

Persistence: this property is related to immutability. In more detail, a ledger satisfies this property when the following holds: a block that has been buried under a sufficient large number of blocks in an honest participant's chain will become eventually part of the ledger with overwhelming probability in the security parameter. This means that all the chains that belong to the honest participants have this block in the same position.

Liveness: a ledger satisfies this property when every transaction issued by an honest participant will eventually get buried deep enough (with overwhelming probability in the security parameter) in order to become a permanent part of the ledger. This guarantees that the adversary cannot censor transactions issued by honest participants.

3.1.2 2-for-1 POWs

In [59] the Bitcoin backbone is used to establish Byzantine agreement [84, 109](we do not provide the background for Byzantine agreement because we do nor refer to it again in the following chapters). However in order for Byzantine agreement to be achieved for honest majority (and not for a ratio of malicious parties at most 1/3) selfish mining attack [48] should be invalidated. The authors of [59] achieve this by making transactions (not only blocks) need computational power to be solved. In a nutshell, each miner performs double proof of work; as happens in Bitcoin each miner computes a hash value and checks if it is smaller than the current target in order to produce a block, but simultaneously it checks if the reverse of the hash value is smaller than the target in order to produce a transaction. In this way, even if the adversary implements selfish mining attack, the honest participants can still try to make their transactions become part of the ledger in the future. This means that the adversary cannot produce a ratio of transactions much higher than its fair ratio.

3.1.3 The Fruitchain Protocol

The above 2-for-1 POW technique is used by [108] that proposes a *fair* blockchain protocol which guarantees the following with overwhelming probability in the security parameter: if we take a large enough segment of an honest chain the fraction of blocks that belongs to any set of honest participants cannot be much smaller compared to its relative computational power.

In a nutshell, each miner mines simultaneously blocks and a set of transactions called *fruits*. In more detail, each miner computes a hash of an input that includes among others (i) the hash of the block it tries to extend (ii) a fingerprint or in other words a hash of the set with the fruits that it wants to include in the ledger and (iii) a nonce.

After that it checks simultaneously if the first κ bits of this hash value is smaller than T_1 in order to produce a block and if the last κ bits of this hash value is smaller than T_2 to produce a fruit. T_2 is much higher than T_1 meaning that it is much easier to produce a fruit. Note that in order for a block to be considered valid it should include fruits that are *recent* meaning that they refer to a recent block. This is important in order to avoid attacks where the adversary pre computes many fruits and tries to include them later in a specific part of the

ledger.

3.2 Game Theoretic Notions

When we study a protocol from a game theoretic perspective we consider the participants as rational or self-interested. This means that when they have to choose among a set of possible actions called *strategies* they choose the strategy that maximizes a quantity that they find important called *utility* (e.g their profit). In the following three chapters of this thesis we present formally the models that we use for our results, but at this point we describe the simplest form of a game called *normal-form game* that allows us to present some basic game theoretic notions that we will use later. This background information is based on [78].

A normal-form game with n participants is the following tuple (S, A, u) where

- S is the set of the participants
- $A = A_1 \times A_2 \ldots \times A_n$ where A_i is the set of all the possible actions/strategies that player i can select. A tuple $\vec{a} = (a_1, \ldots a_n) \in A$ is called *strategy profile* or *joint strategy*.
- $u = (u_1, \ldots, u_n)$ where $u_i : A \to \mathbb{R}$ is the *utility* function of player i. It takes as input a strategy profile and it outputs the value of the quantity that player i wants to maximize (e.g its profit).

Approximate Nash equilibrium: A strategy profile is an ε-*Nash equilibrium* when the following holds for every participant i: if all the participants except i follow the strategy that is indicated by the strategy profile, participant i cannot increase its utility more than ε by choosing a strategy different from the indicated. More formally:

Definition 3 (ε-Nash equilibrium)**.** *A strategy profile $\vec{a} = (a_1, \ldots, a_i, \ldots, a_n) \in A$ is an ε-Nash equilibrium if it satisfies the following property: $\forall i \ \forall a_i' \in A_i$ we have $u_i(\vec{a}) \geq u_i(a_1, \ldots a_i' \ldots, a_n) - \varepsilon$.*

Extended notions of equilibria capture strategic coalitions as well, cf. [7, 12, 22] (see the introduction of Chapter 4 for more details).

Incentive Compatibility: The concept of *Incentive compatibility* appears in a few different forms in the literature. "Dominant-strategy incentive-compatibility" is satisfied when there is not a strictly better strategy than telling the truth or following the protocol respectively whatever the other participants do. "Bayesian-Nash incentive-compatibility" is a weaker notion and a protocol satisfies it when there is a type of Nash equilibrium called "Bayesian Nash equilibrium", where all the participants tell the truth supposing that all the other participants do the same [5]. In cryptocurrency literature some times the incentive compatibility notion is used as equivalent to the Nash equilibrium notion [87]. More broadly, maximizing the profits or maximizing the utility can be seen as an *optimization problem* that includes at least two constraints. The first constraint is *incentive compatibility* and the second constraint is the *participation constraint* which suggests that when a participant participates in the game, this does not result in lower utility compared to not participating [70].

3.3 Chernoff Bound

Let $X_i : i \in \{1, ..., n\}$ be mutually independent Boolean random variables (they can take on two values: 0 or 1; an example is Bernoulli random variables) and $\forall i$ we have $Pr(X_i = 1) = p$. Let $X = \sum_{i=1}^{n} X_i$ and $\mu = pn$. Then we have for any $\delta \in (0, 1]$

$$Pr(X \leq (1-\delta)\mu) \leq e^{-\delta^2 \mu/2}$$

and

$$Pr(X \geq (1+\delta)\mu) \leq e^{-\delta^2 \mu/3}$$

For more detail see eg. [44].

Chapter 4

Incentives in the Bitcoin and the Fruitchain Blockchain Protocol

4.1 Introduction

This chapter is related to the first question this thesis answers and is based on our paper [81]. In more detail, we propose a framework and a notion of Nash equilibrium called *Coalition-Safe Equilibrium with Virtual Payoffs (EVP)* that allows us to study the incentives in blockchain protocols and we use it in order to study incentives in the Bitcoin [59, 100] and the Fruitchain protocol [108].

Recall that a strategy profile is a Nash equilibrium when it satisfies the following property : for every participant i, if all the participants except i follow the indicated strategy (the strategy determined by the strategy profile) then participant i will maximize its utility, if it follows the indicated strategy. In the case of an approximate Nash equilibrium participant i cannot increase *significantly* its utility by choosing a strategy different from the indicated. How much it can gain by deviating determines the approximation in the Nash equilibrium.

This notion can be generalized taking into account coalitions of participants who collaborate and try to maximize the sum of their utilities in the following way: for every coalition C, if all except C follow the indicated strategy, then coalition C cannot increase its joint utility (sum of the utilities of the participants who belong to the coalition) if one or more participants of the coalition choose a strategy different from the indicated. This approach is followed in [108].

In our case we would like to study the incentives in a blockchain protocol by examining if the joint strategy where all the participants follow the protocol is

Nash equilibrium. It is important for a blockchain protocol to satisfy a notion of Nash equilibrium that captures coalitions because then we have the following guarantee: if at some point all the participants follow the protocol, then no coalition will have incentives to deviate, something that could make potentially the ledger that is formulated by this protocol lose some of its desired features such as immutability.

However, the nature of the blockchain protocols demands a more complicated notion compared to Nash equilibrium. In more detail:

1. in most blockchain protocols (such as [100, 108]) the rewards of each participant depend on its blocks in the ledger. As each participant may have a different view on which chain constitutes the (public) ledger, it may have also a different opinion regarding its own rewards and the rewards of each other participant. Thus we characterize the rewards of the participants as *virtual*.

   Moreover, given that the utility of a participant depends on its own rewards and/or the rewards of the other participants, each participant may have a different view on its own utility or the utility of the other participants.

   Note that when we examine a blockchain protocol in the malicious/honest setting, it may satisfy persistence property [59] which means that all the honest participants have common view on the initial part of their chain and thus for the rewards and the utility based on this part.

   However when we study incentives, we consider all the participants as rational so we want our notion to be well defined even if the blockchain protocol we examine does not satisfy persistence or other security properties.

2. in most blockchain protocols the number of blocks that each participant produces and thus its rewards is a probabilistic event.Thus some coalitions may choose a strategy different from the protocol if they know that there is a non negligible probability to increase the quantity they want to maximize (even if the expected value of this quantity if they choose this strategy is not higher). So following the approach of [108], our notion is stronger than Nash equilibrium in this sense and get satisfied by a blockchain protocol only if the following happens: a coalition cannot increase the quantity it wants to maximize with non negligible probability by deviating. To illustrate this the

utility is not defined as the expected value of the quantity the participant wants to maximize over all possible outcomes, as happens usually in game theory [78]. The utility is defined for a specific outcome and reflects the exact value of the quantity that the participant wants to maximize in this outcome.

3. to examine the incentives in a blockchain protocol we need a notion that is well defined for various type of utilities even if they can get negative values. One example is profit.

Our framework takes into account all the above considerations. Moreover we use this framework to study the incentives in the Bitcoin and the Fruitchain protocol [108] according to various utilities.

4.1.1 Our Contribution and Roadmap

In Section 4.3, 4.4, 4.5, 4.6 we describe our framework that includes:

1. the formal definition of reward, cost and utility functions (Section 4.4).

2. an execution model with which we describe a finite execution of a blockchain protocol (Section 4.3. Our model generalizes the execution model of [59] (and is based on the "real-world" protocol execution model of [31, 32, 33, 77]). The additional feature of our model compared to [59] is that certain operations of the protocol are abstracted as oracles and calling such oracles incurs a certain cost to the callee.

3. a notion of Nash equilibrium that takes into account all the above considerations simultaneously and generalizes the notion used in [108] (Section 4.6). In Section 4.2 we compare our notion with [108] and other related notions.

In our results, we revisit three important utility functions: (i) absolute rewards (ii) absolute rewards minus absolute cost and (iii) relative rewards (Section 4.5). With the term *absolute rewards* we refer to the amount of the rewards that a set of participants receives at the end of the execution. With the term *absolute cost* we mean the cost that a set of participants pays during the execution expressed in absolute terms. With the term *relative rewards* we refer to the rewards of a set of participants divided by the total rewards given to all the participants.

For these utility functions, in Section 4.7 we use our framework to prove positive and negative results regarding the incentives in Bitcoin, and we also refer to related works that prove similar results for Bitcoin in their models. In this way we unify previous seemingly divergent views on how the protocol operates in terms of incentives (cf. Theorems 1,2,3,4,5). In a nutshell, these theorems state that Bitcoin with fixed reward per block and with fixed target or target that changes after a fixed number of rounds is an EVP when the utility is absolute rewards or absolute rewards minus absolute cost, while it is not with respect to relative rewards. Section 4.7 is structured in the following way: Firstly we describe our setting and our assumptions and we summarize our notation in a tabular, after that we prove some lemmas that we will need in our proofs, and finally we examine what happens with incentives in Bitcoin for utilities absolute rewards, absolute rewards minus cost and relative rewards when (i) the reward per block is fixed and the target is fixed or changes after a fixed number of rounds or changes when a specific number of blocks are produced (ii) the reward per block is not fixed as it includes also transaction fees.

In Section 4.8 we define a property called "(t, δ)-weak fairness" that is weaker than "fairness" defined in [108], "ideal chain quality" described in [59] and the "race-free property" in [20] (cf. Section 4.8.3 for more details regarding these properties). This allows us to also prove the following result in Section 4.9 regarding incentives in Fruitchain [108]: when the utility is based on relative rewards, the Fruitchain protocol is EVP for coalitions including fewer than half of the number of the participants (Theorem 6,7). In this section we prove also the following novel result regarding incentives of Fruitchain [108]: when the utility is based on absolute rewards minus absolute cost, the Fruitchain protocol is an EVP for coalitions including even up to *all but one* of the participants (Theorem 8). In Section 4.2 we explain in detail how our results are stronger than the results presented in [108], and we refer also to other related works.

In Section 4.9.3 we give an overview of the incentives in the Bitcoin and the Fruitchain protocol.

4.2 Other Related Work

A closely related work that focused on Byzantine Agreement and rational behavior is [66]. Some distinctions between our work and [66] are that (i) their utility

model is tailored to the setting of (single shot) binary Byzantine agreement, while we focus on distributed ledgers that record transaction and rewards for the participants, (ii) in the definition of equilibrium they consider the expectation of utility as opposed to bounds on utility that are supposed to hold with high probability, (iii) at equilibrium, the rational adversary may deviate from the protocol as long as the properties of Byzantine agreement are not violated, while we consider any protocol deviation as potentially invalidating our equilibrium objective as long as the adversarial coalition benefits in the view of one of the other participants.

One model, introduced in [10], that combines Byzantine participants, i.e., participants that can deviate from the protocol arbitrarily, in addition to honest and rational participants, is "BAR." This model includes three types of participants: altruistic, Byzantine and rational and was used to analyse two types of protocols, IC-BFT (Incentive-Compatible Byzantine Fault Tolerant) and Byzantine Altruistic Rational Tolerant (BART) protocols [10]. The first type of protocols (i) satisfies the security properties of a Byzantine Fault Tolerant protocol (safety and liveness) in a setting with Byzantine/honest participants and (ii) guarantees that the best choice for rational participants is to follow the protocol. This guarantee is provided under the following assumptions: (a) if following the protocol is a Nash equilibrium then the rational participants will adopt it as a strategy, (b) rational participants do not collude, and (c) the expected utility of the rational participants is computed considering that the Byzantine participants react in such a way that minimizes the utility of the rational participants. One of the advantages of the IC-BFT model is that it can be used to argue that rational participants have incentives to follow the protocol due to property (ii) and thus they can be considered as honest and in such case the resulting protocol will still be resilient to some Byzantine behaviour due to property (i). Note that [46] had also considered a model including both rational and Byzantine participants with the difference that the Byzantine participants were considered to react arbitrarily and not in such a way that minimizes the utility of the rational participants. The second type of protocols considered in the context of BART guarantee that the security properties are satisfied even if there exist rational participants. So it is weaker than the first type in the sense that it does not guarantee that rational participants will follow the protocol. It guarantees only that even if rational participants deviate in a way that increases their utility, the security properties will not be violated.

Another game theoretic notion that takes into account malicious and rational participants in the context of multi-party computation is called "ε-(k,t)-robust Nash equilibrium" defined in [7]. In this type of equilibrium no participant in a coalition of up to k participants should be able to increase their utility given that there exist up to t malicious participants. As we have already mentioned in our case following [108] when we consider coalitions we study their joint utility (by summing individual rewards) and not the utility of each participant separately something that results in a more relaxed notion in this respect (but still suitable for the distributed ledger setting: following [59, 108] when we study proof of work cryptocurrencies, each participant represents a specific amount of computational power. So a coalition of participants could also be thought to represent one miner).

In [58] a framework for "rational protocol design" is described that is based on the simulation paradigm. That framework was extended and used for examining the incentive compatibility of Bitcoin in [16]. The basic premise is that the miners aim to maximize their expected revenue and the framework describes a game between two participants: a protocol designer D and an attacker A. The Designer D aims to design a protocol that maximizes the expected revenue of the non adversarial participants and keep the blockchain consistent without forks. The adversary A aims to maximize its expected revenue. One difference of our model compared to [16] is that we let the adversary deviate from the protocol not only if its expected utility increases significantly by deviating, but even if it can increase its actual utility significantly just with not negligible probability. In addition [16] focuses exclusively on the incentive compatibility of Bitcoin and only when utility is equivalent to absolute rewards minus absolute cost.

Other related works that study the incentive compatibility of Bitcoin according to a specific utility are [48, 72, 79]. In this chapter we describe how our results compare to these works. In addition, the incentives of nodes who do not want necessarily to engage in mining but they want to use the Bitcoin system for transactions have been studied in [68].

In [108] the Fruitchain protocol is presented, which preserves the security properties of the Bitcoin protocol and also satisfies a property called δ-approximate fairness (assuming honest majority) that prevents the adversary from having a fraction of blocks in an honest chain much higher than its computational power. In addition, in [108] a definition of approximate Nash equilibrium is described,

denoted by"ρ-coalition-safe ε-Nash equilibrium" that guarantees protocol conformity with overwhelming probability. Furthermore, it shows that δ-approximate fairness is enough for ensuring incentive compatibility for coalitions including fewer than half of the participants when the utility is equivalent to absolute rewards.

Our EVP definition is both more general and more explicit in the sense that: (i) it includes a formal description of the properties of the protocol's executions that give rise to the random variables that should be compared. (ii) it includes a formal definition of reward, cost and utility functions. (iii) it takes into account in a rigorous way the fact that local views of honest participants may diverge, and it is well defined even when the underlying protocol does not satisfy persistence.

Moreover our results regarding incentives in the Fruitchain protocol are stronger in the sense that (i) we prove that the Fruitchain protocol is an EVP when the utility is equivalent to absolute rewards for coalitions including even up to all but one participants (not fewer than half of the participants as in [108]) (ii) we prove that the Fruitchain protocol is an EVP also when the utility is equivalent to absolute rewards minus absolute cost (iii) we prove formally what happens in [108] when the utility is equivalent to relative rewards. For more details regarding how our results are compared to the results presented in[108] cf. the related subsections.

As we have already explained we use in one of our proofs a notion of "weak fairness" that we define and is weaker than "fairness" described in [108], "ideal chain quality" described in [59] and "race-free property" in [20]. Another property related to "fairness" is "t-immunity" in [7]. The last property considers utility as an expectation. Note that the notion of fairness has also been used in [90]. A notion of weak fairness has also been used in [92] for a different purpose. Specifically in [92] fairness refers to exchanges between participants; both or neither of the participants take the other's item.

Some other works that investigate the interplay between Cryptography and Game theory in different settings are [6, 7, 65, 75, 106]. In [7, 65] there are also some definitions related to Nash equilibrium that refer also to strategic coalitions, but the utility used is also based on expectation. Some proof-of-stake blockchain protocols that can be proved to be incentive compatible using some notion of equilibrium are [21, 80]. A framework for identifying attacks against the incentive schemes of the blockchain protocols is proposed in [71]. In [25], proof of work

blockchain protocols are modeled as stochastic games while in [93] a survey of game theoretic applications in the blockchain setting is presented.

Previous works on the general topic of rational multi-party protocols include[8, 41, 57, 95, 125] while a related line of research explored cheap talk [7, 61, 88, 121]. For example cheap talk [40, 51] was used in [7] for simulating an honest mediator given (i) secure private channels between agents that incur no cost, (ii) a punishment strategy such as having the participants stop the protocol if misbehaviour is detected.

A game theoretic notion that can be used to handle protocols operating in asynchronous networks is the "ex post Nash equilibrium" and was used in this context in [9, 69]. The way this was used in our context, was to include also adversarial nodes in addition to rational nodes and in [9] the adversarial nodes would determine some specific choices in the protocol execution (such as the initial signal the agents get and the order in which agents are scheduled). The equilibrium condition is required to hold regardless of the choices of the adversarial nodes and even if the rational participants know these choices.

Finally we note that coalition-safety has been examined also in the context of cheap talk [89] and in computational games with mediator [106].

4.3 Our Model

Our model of protocol execution which is an extension to the model used in [59] and is based on [31, 32, 33, 77] consists of the following components:

1. the participants $1, \ldots, n$ that are interactive Turing machine instances (ITIs) or in other words interactive Turing machines (ITM) that run a protocol Π. The set of all the participants is denoted by S.

2. the adversary $\mathcal{A}$ that is an entity that tries to maximize its utility. The adversary is an interactive Turing Machine that is probabilistic and polynomially-bounded (PPT), and it creates a coalition T by corrupting t' participants in the beginning of the execution (the adversary is considered static meaning that the set of the corrupted participants does not change during the execution). Let t be the upper bound on the number of the participants that the adversary can corrupt. This adversarial coalition of the t' participants

will follow the instructions of the adversary instead of running the protocol Π. The set $S \setminus T$ include honest participants who follow the protocol Π.

3. A system of ITM $(\mathcal{Z}, \mathcal{C})$. The first ITM is called *environment* and reflects all the outside world of the protocol. The second ITM is called *control program* and controls the way in which the ITIs and the adversary are allowed to communicate.

The security parameter is denoted by κ.

The model is *hybrid* which means that there exist some Ideal Functionalities that are subroutines of the participants and the adversary. Namely, these subroutines are l oracles $O_1, \ldots, O_l$ and the *Diffuse functionality* defined in [59]. Note that the oracles and the Diffuse functionality are subroutines of all the participants $1, \ldots, n$ (which constitute the set S) and the adversary $\mathcal{A}$. Participants $1, \ldots, n$ and the adversary $\mathcal{A}$ are subroutines of $\mathcal{Z}$.

The Diffuse functionality adjusts the protocol execution in rounds making it synchronous and controls the communication between the honest participants and the adversary. The oracles reflect some cryptographic tools to which the participants have limited access during a round. For example these tools could be a random oracle as in [59] or an oracle providing digital signatures. The limitation in access is controlled by the control program $\mathcal{C}$. The number n of the participants is hardcoded in $\mathcal{C}$, but the participants cannot use it.

The protocol execution is progressing as follows: the environment $\mathcal{Z}$ is forced by $\mathcal{C}$ to spawn the adversary and the participants $1, \ldots, n$. In the beginning of the execution the adversary chooses the set $T = \{i_1, \ldots, i_{t'}\}$ of the t' participants that it will corrupt. Then a round starts and the honest participants are activated by the environment one after other performing a *round-robin* participant execution. Afterwards the adversary is activated. During each round, when an honest participant is activated: (i) it receives via the Diffuse Functionality messages delivered by other participants (for example in Bitcoin the local chains of the other participants) (ii) based on these messages it creates the inputs that it will use for its queries to the oracles (iii) it asks each oracle O_k q_k queries. For each query it incurs cost c_k. After asking the queries, the honest participant sends all the messages/solutions it has produced to the Diffuse Functionality that is responsible for delivering these messages to the other participants, if the adversary permits it. This message/solution in the case of Bitcoin is its local chain. Note that

the adversary can prevent some messages from being delivered to some honest participants before the round ends. When the adversary is activated, it can ask each oracle O_k at most $t' \cdot q_k$ queries. This restriction is achieved via the Control function. Afterwards, the adversary sends its messages to the Diffuse functionality and it specifies the participants to whom it wants its messages to be delivered. After all the honest participants and the adversary have been activated and have sent their messages to the Diffuse Functionality, the round ends. Then, before the beginning of the next round, the Diffuse Functionality delivers to all the honest participants all the messages produced by the other honest participants that were dropped by the adversary during the round .

Note that the Diffuse functionality allows the adversary to deliver first its solution (e.g block) to the honest participants. So when multiple competitive solutions have been produced during a round, the honest participants will probably receive first the adversary's solution. We do not specify the way in which the honest participants decide which message they will give to the Oracles, as this depends on the protocol Π. For example, in the Bitcoin Backbone protocol [59] the honest participants choose to extend the longest chain (thus the input that they give to the random oracle for their queries will be related to this chain) and in the case of a tie they choose the first chain they receive. Thus if the adversary chooses to deliver its solution first, they will adopt the adversarial chain in the case of a tie. For example a different selection rule, such as choosing at random [48], could eliminate the opportunity of the adversary described above in the case of a tie.

In our notion we compare executions performed by arbitrary environments. So we need to fix the number of the rounds the protocol is performed by the environment.

Definition 4. *r-admissible is an environment that performs the blockchain protocol a number of rounds $r = p(\kappa) \neq 0$, where p is a polynomial. When r round ends, the environment terminates the execution. Moreover the input of the environment is $1^{p'(\kappa)}$, where p' is a polynomial.*

4.4 The Reward, Cost and Utility Functions

A protocol Π is associated with a reward and a cost function that allow us to define the utility functions that will be used to examine if a protocol Π satisfies

our notion of Nash equilibrium.

The reward function determines the rewards that a set of participants earns at the end of an execution of the protocol Π. The cost function determines how much cost each individual participant has incurred at the end of an execution due to its queries to the oracles. Recall that the utility function is related to the quantity the participants want to maximize. For example this quantity can be their profit. In our setting the utility function determines the *desired quantity* a set of participants receives at the end of the execution.

As we have explained earlier each participant may have a different view on the (virtual) rewards and thus on the utility of the other participants because the rewards depend on its local view (the messages it has received and the blocks it has produced). Thus the reward and the utility function are parameterised by the participant whose local view we use to determine the rewards and the utility respectively. To be more specific, the reward and the utility function are parameterised by (i) the set of the participants whose rewards and utility we examine and (ii) the participant whose local view determines these rewards and utility. The reward and the utility function get as input an execution and output the rewards and the utility respectively of a set of participants according to the view of an honset participant.

On the other hand, the cost of each participant depends on the number of the queries it asks the oracles during the execution and not on the local view of the other participants. Thus the cost function is parameterised only by the participant whose cost we examine. It takes as input an execution and outputs the cost of this participant. In a proof of work blockchain protocol this cost reflects the cost of the electricity that each miner consumes and/or the cost for having the transactions verified. In a proof of stake protocol this reflects the cost a participant incurs for participating in the protocol. An example is the cost for running a server or verifying transactions.

Note that an execution $\mathcal{E}$ of a protocol can be fully determined by the environment $\mathcal{Z}$, the adversary $\mathcal{A}$, the control program $\mathcal{C}$ and the randomness of these processes and the oracles. We omit the reference to the control program because in all the executions it is the same. The randomness is derived from the participants, the environment, the adversary, and the oracles like the random oracle if they exist as e.g., in [59]. We will use the notation $\mathcal{E}_{\mathcal{Z},\mathcal{A}}$ for the random variable where we have determined the adversary and the environment as algorithms.

Definition 5.

- ***Reward function***: $R_T^j : \mathbb{E} \longrightarrow \mathbb{R}$. *It takes as input an execution* $\mathcal{E} \in \mathbb{E}$ *and outputs the rewards of a set* T *of participants according to the local view of a participant* $j \in S \setminus T$ *at the end of the execution* $\mathcal{E}$.

- $R_T^{\min}(\mathcal{E}_{\mathcal{Z},\mathcal{A}}) = \min\{R_T^j(\mathcal{E}_{\mathcal{Z},\mathcal{A}})\}_{j \in S \setminus T}$. *The* $R^{\min}$ *rewards of a coalition represent the minimum amount of rewards a coalition has received quantified over the view of all* other *participants (which do not belong to the coalition).*

- $R_T^{\max}(\mathcal{E}_{\mathcal{Z},\mathcal{A}}) = \max\{R_T^j(\mathcal{E}_{\mathcal{Z},\mathcal{A}})\}_{j \in S \setminus T}$. *The* $R^{\max}$ *rewards of a coalition represent the maximum amount of rewards a coalition has received quantified over the view of all* other *participants (which do not belong to the coalition).*

- ***Cost function*** : $C_i : \mathbb{E} \longrightarrow \mathbb{R}$. *It has as input an execution* $\mathcal{E} \in \mathbb{E}$ *and outputs the cost a participant* i *has incurred until the end of the execution* $\mathcal{E}$ *due to the queries it asked.*

- ***Utility function***: $U_T^j : \mathbb{E} \longrightarrow \mathbb{R}$. *It has as input an execution* $\mathcal{E} \in \mathbb{E}$ *and outputs the utility of a set* T *of participants according to the local view of a participant* $j \in S \setminus T$ *at the end of the execution* $\mathcal{E}$

- $U_T^{\max}(\mathcal{E}_{\mathcal{Z},\mathcal{A}}) = \max_{j \in S \setminus T}\{U_T^j(\mathcal{E}_{\mathcal{Z},\mathcal{A}})\}$. *The* $U^{\max}$ *rewards of a coalition represent the maximum amount of utility a coalition has quantified over the view of all* other *participants (which do not belong to the coalition).*

- $U_T^{\min}(\mathcal{E}_{\mathcal{Z},\mathcal{A}}) = \min_{j \in S \setminus T}\{U_T^j(\mathcal{E}_{\mathcal{Z},\mathcal{A}})\}$. *The* $U^{\min}$ *rewards of a coalition represent the minimum amount of utility a coalition has quantified over the view of all* other *participants (which do not belong to the coalition).*

4.5 Types of Utilities

We revisit some types of utility that are meaningful for blockchain protocols.

- The first type is based on the absolute rewards of the participants. This means that the participants care about maximizing their actual rewards and ignore their cost. This type was used in [72, 108].

- The second type is based on the absolute rewards minus the absolute cost of the participants. This utility is used when the participants care about maximizing their actual profit. This type was used in [16, 82, 120].

- The third type of utility is based on the relative rewards of the participants and is used when the participants care about earning more money compared to the other participants. This type was used in [20, 48, 79].
- Although we use only three types in our results, we describe also another type of utility that is based on the relative rewards minus the relative cost of the participants. It can be used when the participants care about earning more money compared to the other participants and incurring lower cost compared to the other participants.

Formally

Definition 6. *Let $\mathcal{E} \in \mathbb{E}$ be an arbitrary execution.*

1. **Absolute Rewards.** $U_T^j(\mathcal{E}) = R_T^j(\mathcal{E})$,
2. **Absolute Rewards minus Absolute Cost.** $U_T^j(\mathcal{E}) = R_T^j(\mathcal{E}) - \sum_{l \in T} C_l(\mathcal{E})$,
3. **Relative Rewards.**

$$U_T^j(\mathcal{E}) = \begin{cases} \dfrac{R_T^j(\mathcal{E})}{R_S^j(\mathcal{E})}, & \text{if } R_S^j(\mathcal{E}) \neq 0 \\ 0, & \text{elsewhere} \end{cases}$$

4. **Relative Rewards minus Relative Cost.**

$$U_T^j(\mathcal{E}) = \begin{cases} \dfrac{R_T^j(\mathcal{E})}{R_S^j(\mathcal{E})} - \dfrac{\sum_{l \in T} C_l(\mathcal{E})}{\sum_{l \in S} C_l(\mathcal{E})}, & \text{if } R_S^j(\mathcal{E}), \sum_{l \in S} C_l(\mathcal{E}) \neq 0 \\ 0, & \text{elsewhere} \end{cases}$$

Note that it is possible that no block is produced during the execution and thus $R_S^j(\mathcal{E}) = 0$. In addition if the adversarial coalition includes all the participants and do not ask any questions then $\sum_{l \in S} C_l(\mathcal{E}) = 0$. Thus when we define the utilities that are based on relative rewards and relative costs we take into account this fact.

4.6 Coalition-Safe Equilibria with Virtual Payoffs

Based on these functions (reward, cost and utility functions), we present a formal notion of approximate Nash equilibrium, called c*oalition-safe Equilibrium*

with Virtual Payoffs (EVP). In a nutshell, a protocol Π is an EVP if it guarantees the following with overwhelming probability in the security parameter: a rational strategic actor (hence called the *adversary*) who controls a coalition of participants cannot gain by deviating from the protocol more than an insignificant amount in terms of utility in the view of *any* of the other participants. As a result, for a given protocol Π, if there is a small, but non-negligible, probability that the utility of the adversary when it deviates from Π becomes significantly higher in the view of a single other participant then such protocol will *not be* an EVP.

In more detail, in our notion considering that a rational entity called adversary corrupts a set T of t' participants we examine two executions (recall that t is the upper bound on t'). In both executions the environment that reflects the external world is the same and the participants that are not corrupted by the adversary follow the protocol Π. The executions differ because of the different randomness and the strategy of the adversary.

In the first execution the adversarial coalition follows the protocol Π, or in other words it acts honestly, with the difference that it delivers first its solutions. Recall that according to the Diffuse functionality [59] the adversary has an advantage compared to the honest participants and it can deliver first its solutions to the other participants. This adversarial behaviour is denoted by H_T and the random variable referring to this execution is denoted by $\mathcal{E}_{\mathcal{Z},H_T}$. In [16] a same type of behaviour is called "front running" adversary.

In the second execution denoted by $\mathcal{E}'_{\mathcal{Z},\mathcal{A}}$ the adversary deviates arbitrarily from Π. The two executions are independent random variables, because the first execution does not affect the second execution and vice versa.

The way in which we examine these two executions is by comparing the utilities of the adversary in these two executions for all possible environments. The underlying protocol Π is EVP when with overwhelming probability the $U_T^{\max}$ utility of the adversary when it deviates is not significantly higher compared to its $U_T^{\min}$ utility when it follows the protocol (cf. Section 4.4 for the definitions of $U_T^{\max}$ and $U_T^{\min}$).

Observe that our notion:

1. is related to the Nash equilibrium. Recall that when we examine if a joint strategy is a Nash equilibrium, we assume that all except the coalition follow the protocol and we examine the utility of the coalition. This is similar to

our notion in the sense that we consider all the participants except the coalition honest which means that they follow the protocol.

2. following the approach of [108] considers that an adversarial coalition will deviate only if its joint utility (the sum of the utilities of its members) will increase, not if just one of its member can increase its utility.

   Note that there exist also other notions of Nash equilibria that capture coalitions. Some examples are the following:

   - Strong Nash equilibrium [12]: this notion is satisfied by a joint strategy when no coalition can deviate in a away that increases the utility of *all* its members.
   - "ε-(k,t)-robust equilibrium" defined in [7]: this notion is satisfied by a joint strategy when no member of any coalition can increase its utility when one or more members of the coalition choose a strategy different from the indicated.

   We consider that the approach of examining the sum of the utilities of the coalition is more appropriate for the distributed ledger setting; following [59, 108] when we study proof of work blockchain protocols, each participant represents a specific amount of computational power. Thus a coalition of participants could also be thought as one miner with more computational power.

3. takes into account divergent views of the participants in the following way: the adversary will deviate if there is a non negligible probability to convince just one honest participant that its utility is significantly higher compared to its utility when it follows the protocol.

Definition 7. *Let r be a polynomial in the security parameter κ and $\varepsilon, \varepsilon'$ some small positive constants that are near or equal to zero. The protocol is $(t, \varepsilon, \varepsilon')$-equilibrium with virtual payoffs (EVP) according to a utility $\{U_T^j\}_{j \in S \setminus T}$ when for every PPT (probabilistic and polynomially-bounded) static adversary $\mathcal{A}$ that controls an arbitrary set T including at most t participants and for every r-admissible environment $\mathcal{Z}$, it holds that*

$$U_T^{\max}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) \leq U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) + \varepsilon \cdot \mid U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \mid + \varepsilon'$$

with overwhelming probability in κ.

Note that:

1. this definition is meaningful when the adversary controls a coalition that includes even up to all but one participants, so that $U^{\min}$ and $U^{\max}$ are well defined.

2. $U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T})$ can be negative (if for example is equal to the profit). Thus we use its absolute value.

3. parameters ε and ε' describe the multiplicative and the additive gain respectively in the utility of the adversarial coalition who deviates.

4. this definition captures the concept of *individual rationality* in game theory [79, 99] and *the participation constraint* in optimization problems in economics where the utility of the participants when they participate should not be lower compared to no participating. This happens because one of the arbitrary deviations of the adversary is to abstain from participating.

For simplicity in our results we consider a *static PPT adversary with fixed cost* who decides in the beginning of the execution the number of the queries it will ask. In proof of work blockchains this reflects cloud mining [2].

Definition 8. *A static adversary with fixed cost is an adversary that in the beginning of the execution (i) chooses to corrupt a set* $\boldsymbol{T} = \{i_1, \ldots, i_{t'}\} \subseteq \{1, \ldots, n\} = S$ *of* $t' \leq t$ *participants and (ii) commits to the number of queries each corrupted participant* i_m, $(m = 1, \ldots, t')$ *will ask each oracle* O_k *during each round of the execution. This number is denoted by* $q_k - x_{m,k}$ *(*$x_{m,k}$ *is the number of the queries that* i_m *will not ask* O_k *during each round). It is allowed to choose any strategy, but it is committed to paying during each round the cost that it selected in the beginning of the execution.*

4.7 Incentives in Bitcoin

4.7.1 Setting and Assumptions

When we study incentives in Bitcoin we consider that there is one oracle called random oracle that models a cryptographic hash function (as in [59]). When an honest participant is activated, it adopts the longest chain (and in the case of a

Figure 4.1: Notation in Chapter 4

Notation	
t'	number of participants controlled by the adversary
t	upper bound of t'
r	round after which an execution terminates
n	number of participants
p	probability with which a query to the random oracle gives a block
p_f	probability with which a query to the random oracle gives a fruit
q	number of queries each participant can ask the random oracle during each round
w	flat reward per block (Bitcoin)
w_f	flat reward per fruit (Fruitchain [108])
s	expected number of solutions per round
x	the number of the queries the coalition does not ask during each round
S	the set of all the participants
T	the set of the participants controlled by the adversary
$\overset{text}{=}$	"text" explains why the equality holds

tie the first block it receives) and its asks the random oracle q queries per round trying to extend this chain. It incurs cost c for each query and with probability p it produces a block. Following [59] we assume that if an honest participant produces a block or receives a block via the Diffuse functionality from another participant earlier in the round, it will not try to extend it (for simplicity we assume that it will ask and it will be charged for all its queries but if it finds another block it will drop it). Thus the honest participants extend their local chain by at most one block per round. On the other hand, corrupted participants are allowed to produce more than one block per round. However they can ask the random oracle at most q queries. Recall that our security parameter is denoted by κ.

Rewards: when a participant produces a block, it gets rewarded (according to the view of the participants whose local chains include this block) by a fixed amount w. This means that initially we do not take into account rewards coming from transactions in our analysis. Afterwards, we provide arguments based on previous works regarding what happens if we do so. Recall that transactions give transactions fees to the participants who include them in the block they produce. Additionally a participant can receive rewards if it is recipient in a transaction.

Cost: Each participant incurs cost c for each query that reflects the computational power the participants spend for mining.

Difficulty in mining blocks: when we provide our positive results for incentives in Bitcoin, we consider the following two cases: (i) the difficulty of mining is fixed (ii) the difficulty changes every at least $l \cdot \kappa$ rounds where κ is our security parameter and l a positive constant. In addition, we provide arguments why we cannot have positive results for incentives in Bitcoin when the difficulty changes after a period that depends on the total number of blocks in the chain, as in [100].

Synchronicity: let $s = p \cdot q \cdot n$ be the expected number of solutions per round. Our setting is synchronous which means that s is close to zero (much smaller than 0.1, cf. [59] for more details).

Adversarial model: the adversary is PPT (polynomially-bound and probabilistic Turing machine) static with fixed cost. So it corrupts in the beginning a set

$T = \{i_1, ..., i_{t'}\}$ of $t' \leq t$ participants and decides how many queries each of these participants will ask during each round. Let x_j be the number of the queries that the corrupted participant i_j will not ask during each round. So the queries the adversary will ask during each round will be $t' \cdot q - x$ where $x = \sum_{j=1}^{t'} x_j$.

Utilities: we examine the incentives in Bitcoin considering the first three types of utilities defined in Def. 6. For the first two types (*absolute rewards* and *absolute rewards minus absolute cost*) we provide positive results when (i) the difficulty is fixed or changes after a fixed number of rounds (independent of the number of the blocks that have been produced) and (ii) the reward is fixed, which means that we do not take into account transaction fees. We also argue that if these assumptions do not hold then we cannot have a positive result. For the third type (*relative rewards*) we prove a negative result. Furthermore, we describe how our results compare with the existing results regarding incentives in Bitcoin.

4.7.2 Lemmas

At this point we provide some lemmas we need in order to prove our theorems.

Successful round (defined in [59]) for a subset of participants is a round at the end of which at least one of the participants included in this subset has produced a block.

The following lemma extends *Chain-Growth* Lemma in [59].

Lemma 1. *For every r-admissible environment $\mathcal{Z}$ at the end of each round of an execution $\mathcal{E}_{\mathcal{Z},H_T}$, the local chains of all the honest participants have the same number of blocks that is equal to the successful rounds for all the participants.*

Proof. We prove it by induction on the round r.

Consider an arbitrary execution $\mathcal{E}_{\mathcal{Z},H_T}$. For the basis $r = 1$: if the first round is not a successful round then all the participants have a local chain with length zero equal to the number of the successful rounds which is also zero. If the first round is successful then all the participants have a local chain of length 1. This holds because:

1. the participants cannot have at the end of the first round a local chain with length zero as all the participants at the end of the first round receive from the Diffuse Functionality all the blocks produced during the first round.

Note that H_T follows the protocol and always sends its blocks to the Diffuse functionality.

2. the participants cannot have at the end of the first round a local chain with more than one block given that even if more than one block have been produced during the first round, these blocks can extend the length of the local chains only by one. This holds because (i) if the participants (also the participants that are controlled by H_T) find more than one block they give to the Diffuse functionality only the first block and (ii) even if a participant receives from the Diffuse Functionality in the middle of a round a block produced by another participant, it does not change the block that it tries to extend.

For the induction step we suppose that at the end of the round r all the participants have local chains with length equal to the successful rounds and we can prove with the same arguments that at the end of round $r+1$, if the round $r+1$ is successful, all the participants extend their chain by one block. □

At the end of each round, although the local chains of all the participants have the same length, they may differ in their last part. This happens because there is the case where two or more honest participants have produced a block during the same round.

Lemma 2. *For every r-admissible environment $\mathcal{Z}$, at the end of each round of an execution $\mathcal{E}_{\mathcal{Z},H_T}$, the number of the blocks that are produced by the set T of the participants controlled by the adversary and are included in a local chain of an arbitrary honest participant are equal to the number of the successful rounds for T.*

Proof. Considering an arbitrary execution $\mathcal{E}_{\mathcal{Z},H_T}$ this can be proved by induction on the round r. For the basis $r = 1$: if the first round is not a successful round for T then all the participants have a local chain with zero blocks produced by the participants controlled by the adversary which is equal to the number of the successful rounds for T that is also zero. If the first round is successful for T then all the participants have a local chain that includes exactly one block produced by the adversary H_T. This holds because:

- the participants cannot have at the end of the first round a local chain with no block produced by T, because:

  - H_T follows the protocol and always sends its blocks to the Diffuse functionality which means that all participants receive its blocks.
  - the participants adopt a block produced by T at the end of the first round even if another honest participant has also produced a block during the first round because H_T delivers its blocks first. For simplicity we assume that if there is an honest participant who produces a block in the same round when the adversary produces a block then it also adopts the block produced by the adversary.

- the participants cannot have at the end of the first round a local chain with more than one block produced by T because even if more than one block have been produced by H_T during the first round these blocks can extend the length of the local chains only by one given that H_T follows the protocol.

For the induction step we suppose that at the end of the round r all the honest participants have local chains that include blocks produced by T whose number is equal to the successful rounds for T. Then:

- If round $r+1$ is not a successful round for H_T then the number of the blocks produced by T that are included in the local chain of an arbitrary honest participant does not change.

- If round $r+1$ is successful for T then all the honest participants include exactly one more block produced by T, not necessary the same, because of the arguments described above.

□

By Lemma 2 we can conclude that

Lemma 3. *For every r-admissible environment $\mathcal{Z}$, where κ is the security parameter it holds*

$$R_T^{\max}(\mathcal{E}_{\mathcal{Z},H_T}) = R_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) = X_r^T(\mathcal{E}_{\mathcal{Z},H_T}) \cdot w$$

where $X_r^T(\mathcal{E}_{\mathcal{Z},H_T})$ are the successful rounds for T until the last complete round r of execution $\mathcal{E}_{\mathcal{Z},H_T}$.

Recall w is the reward per block.

Proof. The rewards of T according to the local chain of an honest participant P_j come from the flat reward of each block included in this local chain that is produced by a participant of T. Moreover the flat reward of all the blocks gives the same amount of Bitcoins equal to w. By Lemma 2, at the end of the last complete round r of execution $\mathcal{E}_{\mathcal{Z},H_T}$ all the honest participants have local chains whose number of blocks produced by T is equal to the successful rounds for H_T at the end of the round r. So the maximum reward of T is equal to the minimum reward, as all the local chains of all the honest participants contain the same number of blocks produced by T, and it is equal to the successful rounds for T at the end of round r multiplied by w.

□

4.7.3 Fixed Difficulty in Mining

4.7.3.1 Absolute Rewards

Assuming fixed difficulty in mining we prove that when the adversarial coalition includes even up to all but one participants and the utility is based on absolute rewards (cf. Def.6) Bitcoin is an EVP. Our result is in agreement with the result in [72]. In high level, this holds because (i) the rate with which the adversary produces blocks is independent of the chain it chooses to extend and (ii) when the adversary follows H_T, it reveals always its blocks and thus they get included in the local chains of the honest participants.

Theorem 1. *For any $\delta_1 \in (0, 0.25)$ such that $4 \cdot \delta_1 \cdot (1+s)+s < 1$, where s is the expected number of solutions per round, Bitcoin with fixed target in a synchronous setting with constant reward per block is $(n-1, 4 \cdot \delta_1 \cdot (1+s)+s, 0)$-EVP according to the utility function absolute rewards (cf. Def. 6).*

Proof. Consider $\delta_1 \in (0, 0.25)$ such that $4 \cdot \delta_1 + (1+4 \cdot \delta_1) \cdot s < 1$. We choose also an arbitrary r-admissible environment $\mathcal{Z}$ and an arbitrary adversary $\mathcal{A}$ static with fixed cost that is PPT and it controls an arbitrary set T with t' participants where $t' \in \{1, ..., n-1\}$. Note that when the adversary controls 0 participants then the theorem is proved trivially as the utility of the adversary is zero regardless its strategy.

We examine two executions of the Bitcoin with the same environment, but with different adversary : In the first execution $\mathcal{E}_{\mathcal{Z},H_T}$ the adversary is H_T and

in the second execution $\mathcal{E}'_{\mathcal{Z},\mathcal{A}}$ the adversary is $\mathcal{A}$. Note that the environment is the same in the two executions, which means that it gives the same inputs to the participants and it sends the same messages to the adversary, although it receives different responses from the adversary and specifically it receives no response from H_T. In addition the environment decides before the start of the execution the round $r = p(\kappa) \neq 0$ after which it terminates the protocol. So the two executions last the same number of rounds as they have the same environment.

In more detail, H_T follows protocol and ignores the messages that receives from the environment $\mathcal{Z}$. So in the execution $\mathcal{E}_{\mathcal{Z},H_T}$ the environment can do only the following:

- it gives transactions as input to all the participants.
- it can send messages to H_T, but H_T ignores it.
- it has decided when $\mathcal{E}_{\mathcal{Z},H_T}$ ended.
- it receives outputs from the participants.

Firstly by Lemma 3 and by Chernoff bound we have that:

$$U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \equiv R_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \equiv X_r^T(\mathcal{E}_{\mathcal{Z},H_T}) \cdot w > \frac{p \cdot q \cdot t'}{1 + p \cdot q \cdot t'} \cdot r \cdot (1 - \delta_1) \cdot w > 0 \quad (4.1)$$

with overwhelming probability in r and as $r = p(\kappa)$ also in κ (cf. Section 4.7.1 for the notation). In more detail this can be proved as follows :

- $X^{T,m}(\mathcal{E}_{\mathcal{Z},H_T})$ is a Boolean random variable, where $X^{T,m}(\mathcal{E}_{\mathcal{Z},H_T}) = 1$ when round m was successful for H_T. The variables $\{X^{T,m}(\mathcal{E}_{\mathcal{Z},H_T})\}_{m=1,\ldots,r}$ are independent Bernoulli random variables.
- $X_r^T(\mathcal{E}_{\mathcal{Z},H_T}) \equiv \sum_{m=1}^{r} X^{T,m}(\mathcal{E}_{\mathcal{Z},H_T})$ is the number of the successful rounds for H_T until the last complete round r of $\mathcal{E}_{\mathcal{Z},H_T}$.
- $\forall m \; E[X^{T,m}(\mathcal{E}_{\mathcal{Z},H_T})] = 1 - (1-p)^{qt'}$, where p is the probability with which one query to the random oracle is successful and q is the number of the queries that each participant can ask the oracle during each round. We consider $E[X^{T,m}\mathcal{E}_{\mathcal{Z},H_T})]$ as constant. Note that H_T asks all the available queries.

By Lemma 3 we have that:

$$R_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \equiv R_T^{\max}(\mathcal{E}_{\mathcal{Z},H_T}) \equiv X_r^T(\mathcal{E}_{\mathcal{Z},H_T}) \cdot w \tag{4.2}$$

By Chernoff bound we have that for any $\delta_2 \in (0,1)$ and as a result also for δ_1:

$$Pr[X_r^T(\mathcal{E}_{\mathcal{Z},H_T}) > (1-\delta_1)\cdot(1-(1-p)^{qt'})\cdot r] \geq 1 - e^{-\frac{(\delta_1)^2\cdot(1-(1-p)^{qt'})\cdot r}{2}} \tag{4.3}$$

Recall that q are the queries per round that each honest participant ask the random oracle and p the probability with which each query gives a block.

In addition with probability 1 we prove the following that is stated in [59]:

$$(1-\delta_1)\cdot(1-(1-p)^{qt'})\cdot r \geq \frac{p\cdot q\cdot t'}{1+p\cdot q\cdot t'}\cdot r\cdot(1-\delta_1) \tag{4.4}$$

Specifically, it holds $1-p \leq e^{-p}$ and as a result $1-(1-p)^{q\cdot t'} \geq 1-e^{-p\cdot q\cdot t'}$.

Moreover, we have that $1-e^{-x} \geq x/(1+x)$ for $x \geq 0$ (here $x = p\cdot q\cdot t'$). and as a result :

$$1-(1-p)^{qt'} \geq 1-e^{-p\cdot q\cdot t'} \geq \frac{p\cdot q\cdot t'}{1+p\cdot q\cdot t'}$$

The above inequalities can be proved taking the functions $f(x) = (1-e^{-x})\cdot(1+x)-x$ and $g(x) = e^{-x}-1+x$ for $x \geq 0$ and studying their minimum value using their monotonicity.

So by the above inequality (4.4), by Lemma 3 and by equation (4.3) we conclude equation (4.1) .

In addition for any $\delta \in (0,1)$ and as a result also for δ_1 it holds with overwhelming probability in r and also in κ that:

$$U_T^{\max}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \leq Z_r(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})\cdot w < p\cdot q\cdot t'\cdot r\cdot(1+\delta_1)\cdot w \tag{4.5}$$

where $Z_r(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})$ is the number of the blocks the adversary has produced until the last complete round r of $\mathcal{E'}_{\mathcal{Z},\mathcal{A}}$.

This can be proved with Chernoff bound taking into account the fact that the adversary cannot gain rewards from more blocks than these it has produced. In more detail,

- $Z_{i,j,k}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})$ is a boolean random variable and $Z_{i,j,k}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) = 1$ when at round i of $\mathcal{E'}_{\mathcal{Z},\mathcal{A}}$ the j-th query to the random oracle of the k-th participant controlled by the adversary is successful. $Z_{i,j,k}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})$ are independent Bernoulli random variables.

- $Z_r(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \equiv \sum_{i=1}^{r}\sum_{k=1}^{t'}\sum_{j=1}^{q-x_k} Z_{i,j,k}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})$.

  $$R_T^{\max}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \leq Z_r(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \cdot w$$

  as the adversary cannot gain rewards for more blocks than these it has produced.

- $E[Z_{i,j,k}] = p$

  By Chernoff bound we have that for any $\delta \in (0,1)$ thus also for δ_1

$$Pr[Z_r(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) < p \cdot (q \cdot t' - x) \cdot r \cdot (1+\delta_1)] \geq 1 - e^{-\frac{(\delta_1)^2 \cdot p \cdot (q \cdot t' - x) \cdot r}{3}} \tag{4.6}$$

In addition with probability 1 it holds:

$$p \cdot (q \cdot t' - x) \cdot r \cdot (1+\delta_1) \leq p \cdot q \cdot t' \cdot r \cdot (1+\delta_1)$$

By the above equation we can conclude (4.5).
Finally by equations (4.1),(4.5) we have that

$$U_T^{\max}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \leq U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \cdot (1 + 4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s) \tag{4.7}$$

with overwhelming probability in r and also in κ.

In more detail this can be proved as follows: Let $negl(r)$ be an arbitrary negligible function on r (cf. Chapter 3 for the definition of a negligible function).

- Let F be the final event where it holds

  $$U_T^{\max}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \leq U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \cdot (1 + 4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s)$$

  We want to prove that $Pr[F] \geq 1 - negl(r)$.

- Let A be the event where

  $$U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) > \frac{p \cdot q \cdot t'}{1 + p \cdot q \cdot t'} \cdot r \cdot (1 - \delta_1) \cdot w$$

  By (4.1) we have $Pr[A] \geq 1 - negl(r)$.

- Let B be the event where

  $$U_T^{\max}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) < p \cdot q \cdot t' \cdot r \cdot (1+\delta_1) \cdot w$$

  By (4.5) we have that $Pr[B] \geq 1 - negl(r)$.

- Using the above statements we have that

$$Pr[A \cap B] = Pr[A] - Pr[A \cap \neg B] \geq 1 - negl(r)$$

At this point in order to prove (4.7) we only have to prove that

$$Pr[A \cap B] \leq Pr[F]$$

In order to prove the above statement we suppose that the event $A \cap B$ holds and we prove that the event F holds.

So we have that when $A \cap B$ holds then:

$$\begin{aligned} U_T^{\max}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) &\leq p \cdot q \cdot t' \cdot r \cdot (1+\delta_1) \cdot w \\ &\leq \frac{p \cdot q \cdot t'}{1 + p \cdot q \cdot t'} \cdot r \cdot (1-\delta_1) \cdot w \cdot (1 + 4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s) \\ &< U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \cdot (1 + 4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s) \end{aligned}$$

This means that when $A \cap B$ holds then also F holds that is what we want to prove.

Note that

$$\frac{1+\delta_1}{1-\delta_1} \leq 1 + 4 \cdot \delta_1$$

as $\delta_1 \in (0, 0.25)$. This can be proved if we find the minimum of the function

$$f(x) = -(1+x)/(1-x) + 1 + 4x$$

by its monotonicity.

□

Note that:

- the quantity $4 \cdot \delta_1 \cdot (1+s) + s$ is the multiplicative factor ε that shows how much the adversary can gain if it deviates. Thus the lower s, or in other words the better synchronicity the network has, the better approximate EVP we have.
- recall that the adversary cannot corrupt all the participants so that there is at least one honest participant whose local chain will be used to determine the rewards of the adversary.

4.7.3.2 Absolute Rewards Minus Absolute Cost

If we assume that the cost of each query c is small enough compared to the block reward w, then we can have a similar positive result as above when the utility is based on absolute rewards minus absolute cost . This is in agreement with the result of [16].

Theorem 2. *Suppose that there exists* $\phi \in (0,1-s)$ *such that* $c < p\cdot w\cdot\phi/(1+p\cdot q\cdot(n-1))$. *Then, supposing that the reward of each block is a constant* w, *it holds: for any* $\delta_1 \in (0,0.25)$, *such that* $c \leq p\cdot w\cdot(1-\delta_1)\cdot\phi/(1+p\cdot q\cdot(n-1))$ *and* $4\cdot\delta_1\cdot(1+s)+s<1-\phi$, *where* s *is the expected number of solutions per round, Bitcoin with fixed target in a synchronous setting is* $(n-1,(4\cdot\delta_1\cdot(1+s)+s)/(1-\phi),0)$*-EVP according to the utility function absolute rewards minus absolute cost (cf. Def. 6).*

Proof. We choose an arbitrary r-admissible environment $\mathcal{Z}$ and an arbitrary adversary $\mathcal{A}$ static with fixed cost that is PPT and it controls an arbitrary set T with t' participants where $t' \in \{1,...,n-1\}$. The adversary as described above has chosen the number of the queries that each participant controlled by the adversary does not ask during each round. Let x be the total number of the queries that all the participants controlled by the adversary do not ask during each round.

We have two executions of the Bitcoin protocol with the same environment, but with different adversary : In the first execution $\mathcal{E}_{\mathcal{Z},H_T}$ the adversary is H_T and in the second execution $\mathcal{E}'_{\mathcal{Z},\mathcal{A}}$ the adversary is $\mathcal{A}$.

Consider $\phi \in (0,1-s)$ such that $c < p\cdot w\cdot\phi/(1+p\cdot q\cdot(n-1)) \leq p\cdot w\cdot\phi/(1+p\cdot q\cdot t')$. We choose an arbitrary $\delta_1 \in (0,0.25)$ such that $c \leq p\cdot w\cdot(1-\delta_1)\cdot\phi/(1+p\cdot q\cdot(n-1))$ and $4\cdot\delta_1+(1+4\cdot\delta_1)\cdot s<1-\phi$. Then by hypothesis and by the fact that H_T follows the protocol and asks all the queries that are available to the participants controlled by the adversary during each round we have that :

$$U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) > \frac{p\cdot q\cdot t'}{1+p\cdot q\cdot t'}\cdot r\cdot(1-\delta_1)\cdot w - c\cdot q\cdot t'\cdot r \tag{4.8}$$

$$\geq \frac{p\cdot q\cdot t'}{1+p\cdot q\cdot t'}\cdot r\cdot(1-\delta_1)\cdot w - \frac{q\cdot t'\cdot r\cdot p\cdot w\cdot(1-\delta_1)\cdot\phi}{(1+p\cdot q\cdot t')} \tag{4.9}$$

$$= \frac{p\cdot q\cdot t'}{1+p\cdot q\cdot t'}\cdot r\cdot(1-\delta_1)\cdot w\cdot(1-\phi) \tag{4.10}$$

$$> 0 \tag{4.11}$$

with overwhelming probability in r and also in κ.

Regarding $\mathcal{E}'_{\mathcal{Z},\mathcal{A}}$, given that during each round the adversary asks all the available queries except for the x queries that it has specified in the beginning of the execution, the following holds with overwhelming probability in κ:

$$U_T^{\max}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) < p \cdot (q \cdot t' - x) \cdot r \cdot (1+\delta_1) \cdot w - c \cdot (q \cdot t' - x) \cdot r \tag{4.12}$$

By our assumption that $c \leq p \cdot w \cdot (1-\delta_1) \cdot \phi / (1 + p \cdot q \cdot t')$ for $\phi \in (0, 1-s)$ we have that $f(x) = p \cdot (q \cdot t' - x) \cdot r \cdot (1+\delta_1) \cdot w - c \cdot (q \cdot t' - x) \cdot r$ for $x \in [0, \infty)$ is decreasing.

So it holds with overwhelming probability in κ that:

$$U_T^{\max}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) < p \cdot q \cdot t' \cdot r \cdot (1+\delta_1) \cdot w - c \cdot q \cdot t' \cdot r \tag{4.13}$$

As a result it holds with overwhelming probability in r and in κ

$$\begin{aligned}
&U_T^{\max}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) - U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \\
&\overset{(4.13),(4.8)}{<} p \cdot q \cdot t' \cdot r \cdot (1+\delta_1) \cdot w - \frac{p \cdot q \cdot t'}{1 + p \cdot q \cdot t'} \cdot r \cdot (1-\delta_1) \cdot w \\
&= \frac{p \cdot q \cdot t'}{1 + p \cdot q \cdot t'} \cdot r \cdot (1-\delta_1) \cdot w \cdot (1-\phi) \cdot \left(\frac{(1+\delta_1) \cdot (1 + p \cdot q \cdot t')}{(1-\delta_1) \cdot (1-\phi)} - \frac{1}{1-\phi}\right) \\
&\overset{4.10}{<} U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \cdot \frac{4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s}{1-\phi}
\end{aligned}$$

□

Note that:

- the assumption that there exists $\phi \in (0, 1-s)$ such that $c < p \cdot w \cdot \phi / (1 + p \cdot q \cdot (n-1))$ means that the reward w is high enough compared to the cost of the mining. When the cost is high compared to the rewards and the difficulty of mining not fixed then unexpected behaviours appear (such as using lower computational power) as proved in [54].
- it holds that the smaller the cost of each query is, the better approximation (lower multiplicative factor ε) in EVP we have.

4.7.3.3 Relative rewards

We prove that when the utility is based on relative rewards then following the Bitcoin protocol is not EVP for small multiplicative and additive approximation

factors. Specifically we use the selfish mining strategy described in [34, 48, 63, 102, 115] and we describe a deviation that gives higher utility to the adversary (even if it corrupts one participant) compared to its utility when following H_T. Note that this kind of deviation had been used to prove tightness in *chain quality* in [59].

This result is in agreement with [17, 48] and [108]. However, it has seemingly a contradiction with the result presented in [79] where Bitcoin is modeled as *strategic-release game* and it is proved that honest behaviour is Nash equilibrium if the adversary corrupts a small coalition. This contradiction arises because in [79] the honest participants act as a single identity, something that implies the following: if an honest participant produces a block, then all the other honest participants accept this block temporarily. On the other side in our model in the case of a tie, the block of the adversary wins (recall that the adversary has the ability to deliver its block first and the honest participants accept the first block they receive).

Theorem 3. *Consider* $t \in \{1,...,n-1\}$ *and* $t' < \min\{n/2, t+1\}$. *Then for any* $\varepsilon + \varepsilon' < \frac{t'}{n-t'} \cdot (1-\delta') - \frac{t'}{n} \cdot (1+\delta'') \cdot (1+s)$, *where* s *is the expected number of solutions per round and* δ', δ'' *small positive values, following the Bitcoin with fixed target in a synchronous setting is not* $(t, \varepsilon, \varepsilon')$*-EVP according to the utility function relative rewards (cf. Def.6).*

Proof. Consider $t \in \{1,...,n-1\}$. We consider an arbitrary r-admissible environment $\mathcal{Z}$ and we describe a PPT static adversary A_0 with fixed cost (who controls a set T with $t' < \min\{n/2, t+1\}$ participants) such that it holds with high probability :

$$U_T^{\max}(\mathcal{E}'_{\mathcal{Z},A_0}) - U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \tag{4.14}$$

$$\geq \frac{t' \cdot (1-\delta_1) \cdot (1-\varepsilon''')}{(n-t') \cdot (1+\delta_4)} - \frac{t' \cdot (1+p \cdot q \cdot n)}{n} \cdot \frac{(1+\delta_2)}{(1-\delta_3)} \tag{4.15}$$

$$= B \tag{4.16}$$

for any $\delta_1, \delta_2, \delta_3, \delta_4 \in (0,1)$ and a small $\varepsilon''' > 0$.

After that we prove our theorem by contradiction. In more detail, we suppose that there exist $\varepsilon, \varepsilon'$ such that $\varepsilon + \varepsilon' < B$ so that the Bitcoin with fixed target in asynchronous setting is $(t, \varepsilon, \varepsilon')$-EVP and we end up in contradiction. In other

words we suppose that there exist $\varepsilon, \varepsilon'$ such that $\varepsilon + \varepsilon' < B$ so that

$$U_T^{\max}(\mathcal{E}'_{Z,\mathcal{A}}) - U_T^{\min}(\mathcal{E}_{Z,H_T}) \leq \mid U_T^{\min}(\mathcal{E}_{Z,H_T}) \mid \cdot\varepsilon + \varepsilon' \tag{4.17}$$

with overwhelming probability, where A is an arbitrary PPT static adversary with fixed cost that controls a set T with at most t participants and Z an arbitrary r-admissible environment .

Then we have

$$U_T^{\max}(\mathcal{E}'_{Z,\mathcal{A}}) - U_T^{\min}(\mathcal{E}_{Z,H_T}) \tag{4.18}$$
$$\leq \mid U_T^{\min}(\mathcal{E}_{Z,H_T}) \mid \cdot\varepsilon + \varepsilon' \tag{4.19}$$
$$\leq \varepsilon + \varepsilon' \tag{4.20}$$
$$< B \tag{4.21}$$

with overwhelming probability.

However this does not hold because there exists A_0 that satisfies (4.15).

In order to prove equation (4.15): firstly we find an upper bound for $U_T^{\min}(\mathcal{E}_{Z,H_T})$ that holds with overwhelming probability in the security parameter κ.

Recall that T is an arbitrary set with $t' < \min\{n/2, t+1\}$ participants that adversary A_0, whom we describe later, controls.

By Lemma 2

$$R_T^{\min}(\mathcal{E}_{Z,H_T}) \equiv R_T^{\max}(\mathcal{E}_{Z,H_T}) \equiv X_r^T(\mathcal{E}_{Z,H_T}) \cdot w$$

As a result by Chernoff bound it holds for any $\delta_2 \in (0,1)$ with overwhelming probability in r and also in κ :

$$0 < R_T^{\max}(\mathcal{E}_{Z,H_T}) < p \cdot q \cdot t' \cdot (1+\delta_2) \cdot w \cdot r. \tag{4.22}$$

By Lemma 1

$$R_S^{\min}(\mathcal{E}_{Z,H_T}) \equiv R_S^{\max}(\mathcal{E}_{Z,H_T}) \equiv X_r^S(\mathcal{E}_{Z,H_T}) \cdot w \tag{4.23}$$

As a result for any $\delta_3 \in (0,1)$ with overwhelming probability in r and also in κ :

$$R_S^{\min}(\mathcal{E}_{Z,H_T}) \equiv R_S^{\max}(\mathcal{E}_{Z,H_T}) > \frac{p \cdot q \cdot n}{1 + p \cdot q \cdot n} \cdot r \cdot (1-\delta_3) \cdot w > 0 \tag{4.24}$$

Recall that the executions last at least one round. So we know that with overwhelming probability in κ for any j honest

$$U_T^j(\mathcal{E}_{\mathcal{Z},H_T}) \equiv \frac{R_T^j(\mathcal{E}_{\mathcal{Z},H_T})}{R_S^j(\mathcal{E}_{\mathcal{Z},H_T})}$$

as $R_S^j(\mathcal{E}_{\mathcal{Z},H_T}) \neq 0$.

As a result we have that for any $\delta_2, \delta_3 \in (0,1)$ it holds with overwhelming probability in r and also in κ that:

$$U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \leq U_T^{\max}(\mathcal{E}_{\mathcal{Z},H_T}) \tag{4.25}$$

$$\leq \frac{R_T^{\max}(\mathcal{E}_{\mathcal{Z},H_T})}{R_S^{\min}(\mathcal{E}_{\mathcal{Z},H_T})} \tag{4.26}$$

$$\overset{(4.22),(4.24)}{\leq} \frac{p \cdot q \cdot t' \cdot (1+\delta_2) \cdot w \cdot r}{\frac{p \cdot q \cdot n}{1+p \cdot q \cdot n} \cdot r \cdot (1-\delta_3) \cdot w} \tag{4.27}$$

$$= \frac{t' \cdot (1+p \cdot q \cdot n)}{n} \cdot \frac{(1+\delta_2)}{(1-\delta_3)} \tag{4.28}$$

Now we describe the adversary A_0 who does a type of selfish mining, [34, 48, 63, 102, 115], which was described also in [59] and we find a lower bound for $U_T^{\max}(\mathcal{E}_{\mathcal{Z},A_0})$ with high probability (not negligible).

A_0 chooses to ask all the queries. Initially extends the chain coming from an honest participant, but when it finds a block it does not send it to the Diffuse Functionality. It continues working on its private chain until another participant announces a block. Then the adversary reveals one of its blocks to all the honest participants. When this happens all the honest participants adopt its block instead of the block coming from the honest participant. If the adversarial private chain becomes smaller than the chain coming from an honest participant then the adversary adopts the honest participant's chain. Note that when one of the participants controlled by the adversary finds a block during a round, the adversary uses the rest available queries for finding a block that extends this block.

- $X_r^{S\setminus T}(\mathcal{E}'_{\mathcal{Z},A_0}) \equiv \sum_{m=1}^{r} X^{S\setminus T,m}(\mathcal{E}'_{\mathcal{Z},A_0})$ is the number of the successful rounds for $S \setminus T$ until the last complete round r of $\mathcal{E}'_{\mathcal{Z},A_0}$.
- $Z_r(\mathcal{E}'_{\mathcal{Z},A_0}) \equiv \sum_{i=1}^{r} \sum_{k=1}^{t'} \sum_{j=1}^{q-x_k} Z_{i,j,k}(\mathcal{E}'_{\mathcal{Z},A_0})$. $Z_r(\mathcal{E}'_{\mathcal{Z},A_0})$ is the number of the blocks the adversary A_0 has produced until the last complete round r of $\mathcal{E}'_{\mathcal{Z},A_0}$.

We have that

$$R_S^{\max}(\mathcal{E}'_{\mathcal{Z},A_0}) \equiv R_S^{\min}(\mathcal{E}'_{\mathcal{Z},A_0}) \equiv X_r^{S\setminus T}(\mathcal{E}'_{\mathcal{Z},A_0}) \cdot w \tag{4.29}$$

This holds because of Lemma 1 and due to the fact that the adversary A_0 does not contribute to the extension of the public ledger as it only replaces blocks. In addition it announces its blocks to all honest participants.

In addition with overwhelming probability in κ by Chernoff bound

$$R_S^{\min}(\mathcal{E}'_{\mathcal{Z},A_0}) \equiv R_S^{\max}(\mathcal{E}'_{\mathcal{Z},A_0}) > 0$$

and as a result with overwhelming probability in κ

$$U_T^j(\mathcal{E}_{\mathcal{Z},A_0}) = \frac{R_T^j(\mathcal{E}_{\mathcal{Z},A_0})}{R_S^j(\mathcal{E}_{\mathcal{Z},A_0})}$$

Regarding $R_T^{\min}(\mathcal{E}'_{\mathcal{Z},A_0})$: the adversary A_0 announces its block only if an honest participant finds a block and when this happens, it announces it to all the honest participants. The honest participants always adopt its blocks . So $R_T^{\max}(\mathcal{E}'_{\mathcal{Z},A_0}) \equiv R_T^{\min}(\mathcal{E}'_{\mathcal{Z},A_0})$ with probability 1. The number of the adversarial blocks $B(\mathcal{E}'_{\mathcal{Z},A_0})$ in the local chain of an arbitrary honest at the end of the last complete round r of the execution $\mathcal{E}'_{\mathcal{Z},A_0}$ are, as stated in [59], with high probability equal to the number of the blocks $Z_r(\mathcal{E}'_{\mathcal{Z},A_0})$ produced by the adversary minus a quantity bounded by $\varepsilon'' \cdot p \cdot q \cdot r \cdot t'$, for small $\varepsilon'' > 0$.

This happens because when the adversary A_0 has found more than one block during each round it means that all these blocks form a chain and extend the length of the local chains of all the honest participants. Note that when the execution ends, the adversary may have a small quantity of blocks that are unused in the case the honest participants did not have enough successful rounds.

Recall that contrary to the adversary, when the honest participants have found more than one block during a round, these blocks do not form a chain, because (i) an honest participant never sends more than one block to the Diffuse Functionality, and (ii) when an honest participant receives a block from another participant, it does not extend this new block until the end of the round.

By Chernoff bound we have with high probability for any $\delta_1 \in (0,1)$ and a small $\varepsilon''' > 0$

$$R_T^{\min}(\mathcal{E}'_{\mathcal{Z},A_0}) \geq p \cdot q \cdot t' \cdot r \cdot (1-\delta_1) \cdot w \cdot (1-\varepsilon''') \tag{4.30}$$

Moreover by Chernoff bound it holds with overwhelming probability in κ for any $\delta_4 \in (0,1)$

$$R_S^{\max}(\mathcal{E}'_{\mathcal{Z},A_0}) \leq p \cdot q \cdot (n-t') \cdot r \cdot w \cdot (1+\delta_4) \tag{4.31}$$

So we have with high probability for any $\delta_4, \delta_1 \in (0,1)$ and small $\varepsilon''' > 0$

$$U_T^{\max}(\mathcal{E}'_{\mathcal{Z},A_0}) \geq U_T^{\min}(\mathcal{E}'_{\mathcal{Z},A_0}) \tag{4.32}$$

$$\geq \frac{R_T^{\min}(\mathcal{E}'_{\mathcal{Z},A_0})}{R_S^{\max}(\mathcal{E}'_{\mathcal{Z},A_0})} \tag{4.33}$$

$$\geq \frac{p \cdot q \cdot t' \cdot r \cdot (1-\delta_1) \cdot (1-\varepsilon''') \cdot w}{p \cdot q \cdot (n-t') \cdot r \cdot w \cdot (1+\delta_4)} \tag{4.34}$$

$$= \frac{t' \cdot (1-\delta_1) \cdot (1-\varepsilon''')}{(n-t') \cdot (1+\delta_4)} \tag{4.35}$$

Finally by the above and by equality (4.25) for any $\delta_1, \delta_2, \delta_4, \delta_3 \in (0,1)$ and small $\varepsilon''' > 0$ it holds with high probability :

$$U_T^{\max}(\mathcal{E}'_{\mathcal{Z},A_0}) - U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \tag{4.36}$$

$$\geq \frac{t' \cdot (1-\delta_1) \cdot (1-\varepsilon''')}{(n-t') \cdot (1+\delta_4)} - \frac{t' \cdot (1+p \cdot q \cdot n)}{n} \cdot \frac{(1+\delta_2)}{(1-\delta_3)} \tag{4.37}$$

$$= B \tag{4.38}$$

□

4.7.4 Difficulty in Mining Changes After a Fixed Number of Rounds

In the following theorem we extend the above positive results when the difficulty in mining changes after fixed periods that consist of at least $l \cdot \kappa$ rounds where l is a positive constant and κ the security parameter. With fixed periods we mean periods that are independent of the number of the blocks that have been produced.

4.7.4.1 Absolute Rewards

Theorem 4. *Supposing that (i) the block reward changes every at least $l \cdot \kappa$ rounds where l is a positive constant and κ the security parameter and (ii) the environment terminates the execution at least $l \cdot \kappa$ rounds after the last change of the block reward then it holds: for any $\delta_1 \in (0, 0.25)$ such that $4 \cdot \delta_1 \cdot (1+s) + s < 1$, where s is the expected number of solutions per round, Bitcoin with fixed target in a synchronous setting is $(n-1, 4 \cdot \delta_1 \cdot (1+s) + s, 0)$-EVP according to the utility function absolute rewards (cf. Def.6).*

Intuitively the above theorem holds because the number of blocks that are produced in a round does not depend on the number of blocks produced in the previous round. So we can use the previous theorem for each single period where the difficulty of mining is fixed.

Proof. By Lemma 2 we have the following lemma.

Lemma 4. *For every r-admissible environment $\mathcal{Z}$ with input $1^{p'(\kappa)}$, where κ is the security parameter it holds*

$$R_T^{\max}(\mathcal{E}_{\mathcal{Z},H_T}) = R_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) = \sum_{j=1}^{m+2} X_{r_j}^T(\mathcal{E}_{\mathcal{Z},H_T}) \cdot w_{j-1}$$

where $r_1,...,r_m$ are the rounds when the block reward changes, r_0 is the first round, r_{m+1} the last complete round of execution $\mathcal{E}_{\mathcal{Z},H_T}$, $r_{m+2} = r_{m+1}+1$, $w_0, w_1, ..., w_m = w_{m+1}$ are the block rewards respectively and $X_{r_j}^T(\mathcal{E}_{\mathcal{Z},H_T})$ are the successful rounds for T between the rounds r_{j-1} and $r_j - 1$ included r_{j-1} and $r_j - 1$.

Note that $X_{r_j}^T(\mathcal{E}_{\mathcal{Z},H_T})$ is a sum of independent Boolean random variables that are Bernoulli random variables. In addition

Lemma 5.

$$R_T^{\max}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) \leq \sum_{j=1}^{m+2} Z_{r_j}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) \cdot w_{j-1},$$

where $r_1,...,r_m$ are the rounds when the block reward changes, r_0 is the first round, r_{m+1} the last complete round of execution $\mathcal{E}_{\mathcal{Z},H_T}$, $r_{m+2} = r_{m+1}+1$, $w_0, w_1, ..., w_m = w_{m+1}$ are the block rewards respectively, $Z_{r_j}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}})$ is the number of blocks produced by the adversary between the rounds r_{j-1} and $r_j - 1$ included r_{j-1} and $r_j - 1$.

Furthermore We have for $j \in \{1,...,m\}$ for any $\delta_2 \in (0,1)$

$$X_{r_j}^T(\mathcal{E}_{\mathcal{Z},H_T}) \cdot w_j > \frac{p \cdot q \cdot t'}{1 + p \cdot q \cdot t'} \cdot (r_j - r_{j-1}) \cdot (1 - \delta_2) \cdot w_j > 0$$

with overwhelming probability in $r_j - r_{j-1}$ and as $r_j - r_{j-1} \geq l \cdot \kappa$ also in κ.

In addition

$$(X_{r_{m+1}}^T(\mathcal{E}_{\mathcal{Z},H_T}) + X_{r_{m+2}}^T(\mathcal{E}_{\mathcal{Z},H_T})) \cdot w_m > \frac{p \cdot q \cdot t'}{1 + p \cdot q \cdot t'} \cdot (r_{m+1} - r_m + 1) \cdot (1 - \delta_2) \cdot w_m$$

with overwhelming probability in $r_{m+1} - r_m + 1 \geq l \cdot \kappa$.

Moreover for $j \in \{1,...,m\}$ for any $\delta_1 \in (0,1)$ it holds

$$Z_{r_j}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) < p \cdot q \cdot t' \cdot (r_j - r_{j-1}) \cdot (1 + \delta_1)$$

with overwhelming probability in $r_j - r_{j-1}$ and as $r_j - r_{j-1} \geq l \cdot \kappa$ also in κ.

Also

$$Z_{r_{m+1}}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) + Z_{r_{m+2}}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) < p \cdot q \cdot t' \cdot (r_{m+1} - r_m + 1) \cdot (1 + \delta_1)$$

with overwhelming probability in $r_{m+1} - r_m + 1 \geq l \cdot \kappa$.

So for $j \in \{1, ..., m\}$ it holds for any $\delta_1 \in (0, 0.25)$

$$Z_{r_j}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) \cdot w_j < X^T_{r_j}(\mathcal{E}_{\mathcal{Z},H_T}) \cdot w_j \cdot (1 + 4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s)$$

with overwhelming probability in κ.

In addition

$$(Z_{r_{m+1}}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) + Z_{r_{m+2}}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}})) \cdot w_m < (X^T_{r_{m+1}}(\mathcal{E}_{\mathcal{Z},H_T}) + X^T_{r_{m+2}}(\mathcal{E}_{\mathcal{Z},H_T})) \cdot w_m \cdot l$$

with overwhelming probability in κ, where $l = (1 + 4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s)$.

As a result with overwhelming probability in κ it holds for any $\delta_1 \in (0, 0.25)$

$$\begin{aligned}
&R^{\max}_T(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) \\
&\leq \sum_{j=1}^{m} [Z_{r_j}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) \cdot w_{j-1}] + (Z_{r_{m+1}}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) + Z_{r_{m+2}}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}})) \cdot w_m \\
&< \sum_{j=1}^{m} [X^T_{r_j}(\mathcal{E}_{\mathcal{Z},H_T}) \cdot w_{j-1} \cdot (1 + 4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s)] \\
&+ (X^T_{r_{m+1}}(\mathcal{E}_{\mathcal{Z},H_T}) + X^T_{r_{m+2}}(\mathcal{E}_{\mathcal{Z},H_T})) \cdot w_m \cdot (1 + 4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s) \\
&\overset{w_m = w_{m+1}}{\leq} (\sum_{j=1}^{m+2} X^T_{r_j}(\mathcal{E}_{\mathcal{Z},H_T}) \cdot w_{j-1}) \cdot (1 + 4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s) \\
&\equiv R^{\min}_T(\mathcal{E}_{\mathcal{Z},H_T}) \cdot (1 + 4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s)
\end{aligned}$$

□

4.7.4.2 Absolute Rewards Minus Absolute Cost

Theorem 5. *Assume that (i) the block reward changes every at least $l \cdot \kappa$ rounds where l is a positive constant and κ the security parameter and (ii) the environment terminates the execution at least $l \cdot \kappa$ rounds after the last change of the block reward. Let w_j for $j \in \{0, ..., m\}$ be all the block rewards respectively for each participant. Assuming that there exists $\phi \in (0, 1-s)$ such that $c < p \cdot w_j \cdot \phi / (1 + p \cdot q \cdot (n-1))$ for all $j \in \{0, ..., m\}$, then it holds: for any $\delta_1 \in (0, 0.25)$, such that $c \leq$*

$p \cdot w_j \cdot (1-\delta_1) \cdot \phi/(1+p \cdot q \cdot (n-1))$ *for all* $j \in \{0,...,m\}$ *and* $4 \cdot \delta_1 \cdot (1+s)+s < 1-\phi$, *where* s *is the expected number of solutions per round, Bitcoin with fixed target in a synchronous setting is* $(n-1,(4 \cdot \delta_1 \cdot (1+s)+s)/(1-\phi),0)$*-EVP according to the utility function absolute rewards minus absolute cost (cf. Def. 6).*

Proof. By Lemmas 4,5 and by the fact that the adversary is static with fixed cost we have the following lemmas

Lemma 6. *For every r-admissible environment* $\mathcal{Z}$ *with input* $1^{p'(\kappa)}$, *where* κ *is the security parameter it holds*

$$U_T^{\max}(\mathcal{E}_{\mathcal{Z},H_T}) = U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) = \sum_{j=1}^{m+2} X_{r_j}^T(\mathcal{E}_{\mathcal{Z},H_T}) \cdot w_{j-1} - \sum_{j=1}^{m+2} \sum_{l:P_l \in T} C_{l,r_j}(\mathcal{E}_{\mathcal{Z},H_T})$$

where $r_1,...,r_m$ *are the rounds when the block reward changes,* r_0 *is the first round,* r_{m+1} *the last complete round of execution* $\mathcal{E}_{\mathcal{Z},H_T}$, $r_{m+2} = r_{m+1}+1$, $w_0,w_1,...,w_m = w_{m+1}$ *are the block rewards respectively,* $X_{r_j}^T(\mathcal{E}_{\mathcal{Z},H_T})$ *are the successful rounds for* T *and* $\sum_{l:P_l \in T} C_{l,r_j}(\mathcal{E}_{\mathcal{Z},H_T})$ *the cost for* T *respectively between the rounds* r_{j-1} *and* $r_j - 1$ *included* r_{j-1} *and* $r_j - 1$.

Recall that the cost of each round is fixed and determined in the beginning of the execution.

Lemma 7.

$$U_T^{\max}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \leq \sum_{j=1}^{m+2} Z_{r_j}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \cdot w_{j-1} - \sum_{j=1}^{m+2} \sum_{l:P_l \in T} C_{l,r_j}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \tag{4.39}$$

where $r_1,...,r_m$ *are the rounds when the block reward changes,* r_0 *is the first round,* r_{m+1} *the last complete round of execution* $\mathcal{E'}_{\mathcal{Z},\mathcal{A}}$, $r_{m+2} = r_{m+1}+1$, $w_0,w_1,...,w_m = w_{m+1}$ *are the block rewards respectively,* $Z_{r_j}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})$ *is the number of blocks produced by the adversary between the rounds* r_{j-1} *and* $r_j - 1$ *included* r_{j-1} *and* $r_j - 1$.

Consider $t' \in \{1,...,n-1\}$, $\phi \in (0,1-s)$ such that $c < p \cdot w_j \cdot \phi/(1+p \cdot q \cdot (n-1)) \leq p \cdot w_j \cdot \phi/(1+p \cdot q \cdot t')$ for all $j \in \{0,...,m\}$ and $\delta_1 \in (0,0.25)$ such that $c \leq p \cdot w_j \cdot (1-\delta_1) \cdot \phi/(1+p \cdot q \cdot (n-1))$ for all $j \in \{0,...,m\}$ and $4 \cdot \delta_1 + (1+4 \cdot \delta_1) \cdot s < 1-\phi$.

By the previous lemmas, by the assumption that for any $j \in \{0,...,m\}$, $c \leq p \cdot w_j \cdot (1-\delta_1) \cdot \phi/(1+p \cdot q \cdot t')$ and by Chernoff bound we have with overwhelming probability in κ :

$$U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \equiv \sum_{j=1}^{m}[X_{r_j}^T(\mathcal{E}_{\mathcal{Z},H_T})\cdot w_{j-1}] + (X_{r_{m+1}}^T(\mathcal{E}_{\mathcal{Z},H_T}) + X_{r_{m+2}}^T(\mathcal{E}_{\mathcal{Z},H_T}))\cdot w_m -$$
$$\sum_{j=1}^{m+2}[\sum_{l\in T} C_{l,r_j}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})]$$
$$> \sum_{j=1}^{m}[\frac{p\cdot q\cdot t'}{1+p\cdot q\cdot t'}\cdot(r_j - r_{j-1})\cdot(1-\delta_1)\cdot w_{j-1}\cdot(1-\phi)] +$$
$$\frac{p\cdot q\cdot t'}{1+p\cdot q\cdot t'}\cdot(r_{m+1}-r_m+1)\cdot(1-\delta_1)\cdot w_m\cdot(1-\phi)$$
$$> 0$$

and

$$U_T^{\max}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \leq \sum_{j=1}^{m}[Z_{r_j}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})\cdot w_{j-1}] + (Z_{r_{m+1}}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) + Z_{r_{m+2}}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}))\cdot w_m -$$
$$\sum_{j=1}^{m+2}\sum_{l\in T} C_{l,r_j}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})$$
$$< \sum_{j=1}^{m}[p\cdot(q\cdot t')\cdot(r_j - r_{j-1})\cdot(1+\delta_1)\cdot w_{j-1} - c\cdot(q\cdot t')\cdot(r_j - r_{j-1})] +$$
$$p\cdot(q\cdot t')\cdot(r_{m+1}-r_m+1)\cdot(1+\delta_1)\cdot w_m - c\cdot(q\cdot t')\cdot(r_{m+1}-r_m+1)$$

As a result, we can prove in the same way as the previous subsection that it holds with overwhelming probability in r and in κ the following :

$$U_T^{\max}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) - U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \leq U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T})\cdot\frac{4\cdot\delta_1 + (1+4\cdot\delta_1)\cdot s}{1-\phi}$$

□

4.7.5 Difficulty Changes After a Number of Rounds That Depends on the Number of Blocks in the Chain

4.7.5.1 Absolute Rewards

If we consider that the difficulty of mining changes as specified in [100] then Bitcoin is not an EVP with good approximation (low $\varepsilon, \varepsilon'$).

Recall that according to [100] the difficulty in mining changes when a specific number of blocks has been included in the chain. This means that the period after which the difficulty changes depends on the number of the blocks in the chain.

Specifically it is adapted so that every 10 minutes one block in expectation is produced. In more detail, if fewer blocks than the expected have been produced then the difficulty is reduced, and if more blocks than expected are produced the opposite. This tries to incorporate the fact that the total computational power that is used for mining each period can differ.

In this case following the Bitcoin protocol is not an EVP with good approximation because the adversary can increase its utility by deviating in the following way: it implements selfish mining attack [48] in the first rounds and follows $\boldsymbol{H_T}$ in the following rounds. This increases its utility because when it implements selfish mining attack, the difficulty is reduced (see background for more details), because the total number of blocks are fewer than the expected based on the real total computational power. In this way it can produce more blocks and get more rewards in the following rounds (see also [67]). In [67] they also propose a different selection rule for adjusting the target that takes into account also the blocks in the forks.

4.7.6 When Transactions Contribute to the Rewards

In our analysis until now we assume that each block gives a fixed reward. But what does it happen when the rewards come also from the transactions included in the blocks?

At this point following [16, 59] we consider that the inputs which the environment gives to the participants are transactions. A transaction can contribute to the rewards of a participant in the following two ways: i) it gives transaction fees to the participant who includes it in the block it produces ii) it gives the declared amount to the recipient of the transaction. These extra sources of rewards open the door to deviations that could give to the adversary higher utility compared to following $\boldsymbol{H_T}$. Some examples are some attacks described in [26, 91]. In our setting these attacks reflect the following scenario: the environment makes some of the corrupted participants recipients of an amount of Bitcoins in some transactions that are invalid in the longest chain and valid in a smaller chain. Thus the adversary may have incentives to deviate by extending the smaller chain. These observations agree with [16] that describes some distributions of inputs which make Bitcoin not incentive compatible.

4.8 Incentives in a Fair Blockchain Protocol

4.8.1 (t,δ)-weak fairness

In this section we describe a property which is sufficient for proving that a protocol is EVP when the utility is based on relative rewards (cf. Def.6). This property is called "(t,δ)-weak fairness" and can aid in the design of EVP protocols.

A protocol will satisfy "(t,δ)-weak fairness" property when with overwhelming probability in the security parameter:

1. the fraction of the rewards the set $S \setminus T$ earns (the set of all the honest participants) is lower bounded by $(1-\delta)$ times its relative cost even if the adversarial coalition including at most t participants deviates.

2. if the adversary follows H_T program, any set of participants gets a fraction of rewards lower bounded by $(1-\delta)$ times its relative cost.

Formally:

Definition 9. *A blockchain protocol satisfies (t,δ)-weak fairness if for any r-admissible environment $\mathcal{Z}$, for any PPT adversary $\mathcal{A}$ which controls a set T with at most t participants and for any $j \in S \setminus T$, where S is the set of all the participants, we have with overwhelming probability in the security parameter κ:*

- $R^j_{S\setminus T}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) \geq (1-\delta) \cdot \frac{\sum_{l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} \cdot R^j_S(\mathcal{E}'_{\mathcal{Z},\mathcal{A}})$

- *for any subset $S_H \subseteq S$ it holds* $R^j_{S_H}(\mathcal{E}_{\mathcal{Z},H_T}) \geq (1-\delta) \cdot \frac{\sum_{l\in S_H} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} \cdot R^j_S(\mathcal{E}_{\mathcal{Z},H_T})$ *where* $\delta \in [0,1)$.

Note that:

- the quantity $\sum_{l\in S_H} C_l(\mathcal{E}_{\mathcal{Z},H_T}) / \sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})$ reflects S_H's relative computational power.

- given that during each round H_T and honest participants follow the Bitcoin protocol and thus they ask all the available queries (which are q per round per participant, cf. Section 4.7.1) it holds $\sum_{l\in S_H} C_l(\mathcal{E}_{\mathcal{Z},H_T}) / \sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T}) = (c\cdot q\cdot r\cdot t_H)/(c\cdot q\cdot r\cdot n)$ where t_H is the number of participants of S_H.

- the environment performs an execution of at least one round so $\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T}) \neq 0$, but still we cannot exclude a scenario where no block is produced and $R^j_S(\mathcal{E}_{\mathcal{Z},H_T})$ is zero. Thus we do not divide with this quantity.

4.8.2 Using (t,δ)-weak fairness for Designing EVP Protocols

The following theorem states that if (i) a protocol satisfies (t,δ)-weak fairness property and (ii) with overwhelming probability the total rewards are not zero, then following the protocol is EVP under an adversarial coalition with at most t participants when the utility is based on relative rewards. We use this theorem to prove that the Fruitchain protocol [108] is EVP when the utility is based on relative rewards.

Theorem 6. *When a protocol satisfies (t,δ)-weak fairness and in addition for any $j \in S \setminus T$, for any PPT adversary $\mathcal{A}$ which controls a set T with at most t participants and for any r-admissible environment $\mathcal{Z}$ it holds $R_S^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) > 0$ with overwhelming probability in the security parameter κ, then following the protocol is $(t,0,\delta)$-EVP according to the utility function relative rewards (cf. Def.6).*

Proof. We choose an arbitrary r-admissible environment $\mathcal{Z}$, and an arbitrary adversary $\mathcal{A}$ static that is PPT and it controls a set T that it includes $t' \leq t$ participants. We examine two executions of the blockchain protocol with the same environment $\mathcal{Z}$, but with different adversary: In the first execution $\mathcal{E}_{\mathcal{Z},H_T}$ the adversary is H_T and in the second execution $\mathcal{E'}_{\mathcal{Z},\mathcal{A}}$ the adversary is $\mathcal{A}$.

We prove that with overwhelming probability in the security parameter for any j honest we have:

$$U_T^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \equiv \frac{R_T^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})}{R_S^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})} \leq \frac{\sum_{l\in T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} + \delta \cdot \frac{\sum_{l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} \tag{4.40}$$

By (t,δ)-weak fairness and by the fact that for any j honest it holds with overwhelming probability $R_S^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) > 0$ we have the following result:

for any j honest it holds with overwhelming probability in the security parameter

$$\begin{aligned}
&R_{S\setminus T}^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \geq (1-\delta)\cdot \frac{\sum_{l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} \cdot R_S^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \Rightarrow \\
&R_T^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \leq R_S^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \cdot (1-(1-\delta)\cdot \frac{\sum_{l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})}) \Rightarrow \\
&R_T^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \leq R_S^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \cdot (\frac{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} - (1-\delta)\cdot \frac{\sum_{l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})}) \Rightarrow \\
&\frac{R_T^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})}{R_S^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})} \leq \frac{\sum_{l\in T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} + \delta \cdot \frac{\sum_{l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})}
\end{aligned}$$

Note that with overwhelming probability $R_S^j(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) > 0$ and as a result

$$U_T^j(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) \equiv \frac{R_T^j(\mathcal{E}'_{\mathcal{Z},\mathcal{A}})}{R_S^j(\mathcal{E}'_{\mathcal{Z},\mathcal{A}})} \tag{4.41}$$

By weak fairness and by the fact that it holds with overwhelming probability $R_S^{\min}(\mathcal{E}_{\mathcal{Z},\mathcal{H}_T}) > 0$ we have the following result:

$$U_T^{min}(\mathcal{E}_{\mathcal{Z},H_T}) \geq (1-\delta) \cdot \frac{\sum_{l\in T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} \tag{4.42}$$

By equations (4.40), (4.42) we have that with overwhelming probability in the security parameter

$$\begin{aligned} U_T^{\max}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) - U_T^{min}(\mathcal{E}_{\mathcal{Z},H_T}) \leq & \frac{\sum_{l\in T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} + \\ & \delta \cdot \frac{\sum_{l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} - (1-\delta) \cdot \frac{\sum_{l\in T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} \\ & \leq \delta \end{aligned}$$

□

4.8.3 Comparison Between (t,δ)-weak fairness and Other Notions

Our property is weaker than (T,δ)-*approximate fairness w.r.t.* ρ *attackers* defined in [108] and *ideal chain quality* defined in [59].

The property (T,δ)-*approximate fairness w.r.t.* ρ *attackers* defined in [108] says that in any sufficient long window of the chain with T blocks, any set of honest participants with computational power ϕ will have with overwhelming probability at least $(1-\delta)\cdot\phi$ fraction of the blocks regardless what the adversary with a fraction of computational power at most ρ does.

Ideal chain quality defined in [59] says that any coalition of participants (regardless the mining strategy they follow) will have a fraction of blocks in the blockchain that is proportional to their collective relative hashing power.

Let us assume that one of the following holds:

1. each participant is rewarded with a fixed amount for each block it has produced in the chain.

2. there is a reward pot that is shared evenly among the producers of blocks in the chain at the end of the execution and each block contributes to this reward pot an amount that is not fixed (because for example of the transaction fees).

Then our definition is weaker than (T, δ)-*approximate fairness w.r.t.* ρ *attackers* defined in [108], *ideal chain quality* defined in [59] and *race-free property* defined in [20] in the following aspect: in the case when the adversarial coalition deviates from H_T program, our property does not guarantee that any subset of honest participants earn a fraction of rewards at least $(1-\delta)$ times its relative cost. It guarantees this only for the whole set of honest participants [1].

This means that our definition allows the adversary to reduce the fraction of the rewards of a specific honest participant if the fraction of the rewards of all the honest participants is not affected in a significant way.

4.9 Incentives in Fruitchain

In this section we analyze the incentives in the Fruitchain protocol [108]. Firstly we decribe an abstraction of the Fruitchain protocol that is compatible with our model. We use notation from both [59] and [108].

In the Fruitchain protocol [108] there exists just one oracle: the random oracle that models a cryptographic hash function. The participants are activated in a "round-robin" way. When a participant is activated, it follows the same procedure as Bitcoin with the following main diferences:

- in the input that it is going to give to the random oracle it includes "a fingerprint" of all the fruits that refer to a recent block (as defined in [108]) and they have not been included in the blockchain yet.
- when it asks the random oracle a query, it produces with probability p a block and with probability p_f a fruit, where $p_f >> p$. Note that the difficulty for mining both the fruits and the blocks is fixed.
- an honest participant is allowed to produce more than one fruits during each round but still at most one block.

[1] To be more accurate we can say that our definition is weaker if the environment is restricted to perform the protocol a number of rounds that are enough so that with overwhelming probability the local chain of any honest participant has length at least T. We need this restriction because T has not been used in our definition as parameter.

The number of the participants is denoted by n and the upper bound on the queries asked by each participant during each round is denoted by q. Recall that S is the set of all the participants and T the set of the corrupted participants controlled by the adversary.

In our results we consider that each participant incurs cost c for each query to the random oracle. Moreover we assume that: (i) the adversary is static which means that decides in the beginning who participants it corrupts (ii) the model is synchronous (at the end of each round each participant receives all the messages sent by the other honest participants) (iii) only the fruits give a reward that is considered fixed and is denoted by w_f [2].

4.9.1 Relative Rewards

The following theorem states that the Fruitchain protocol is EVP (with only additive approximation factor, not multiplicative) when the adversarial coalition includes fewer than half of the participants and the utility is based on relative rewards (cf. Def. 6).

This theorem shows formally that simultaneous mining of blocks and fruits used in [108] [3] does not allow an adversarial coalition to perform selfish mining [48] and increase significantly its relative rewards.

Theorem 7. *Consider* $\delta \in (0,1)$ *and* T_0 *such that the Fruitchain protocol satisfies* (T_0, δ)*-approximate fairness property. Then the Fruitchain protocol is* $(n/2-1, 0, \delta)$*-EVP according to the utility function relative rewards (cf. Def. 6), under an* r*-admissible environment where* $r \geq T_0/(p_f \cdot (\frac{n}{2}+1) \cdot (1-\delta) \cdot q)$.

Proof. Given that the Fruitchain protocol satisfies (T_0, δ)-approximate fairness property when the adversary controls at most $n/2-1$ participants, then it satisfies also $(n/2-1, \delta)$-weak fairness property under the restriction that the environment performs the protocol so many rounds that with overwhelming probability (in the security parameter) any honest participant has a chain of at least T_0 fruits. Note that by chain growth rate proved in [108] when $r \geq \frac{T_0}{p_f \cdot (\frac{n}{2}+1) \cdot (1-\delta) \cdot q}$ and

[2] We consider that only the fruits give a reward because we want to use the *fairness* property that is satisfied by the Fruitchain protocol [108]. Note that the fainess property refers to the fruits, because actual blocks are possibly still vulnerable to selfish mining attacks [48].

[3] This technique is based on the 2-for-1 POW technique that was was initially proposed for the mitigation of selfish mining in [59] in the context of achieving Byzantine agreement for honest majority.

the adversary controls at most $n/2-1$ participants, then indeed it holds that with overwhelming probability any honest participant has a chain of at least T_0 fruits. In addition, by Chernoff bound and by the fact that the execution lasts at least one round, it holds with overwhelming probability in κ the following: for any j honest, for any PPT static adversary $\mathcal{A}$ that controls at most $n/2-1$ participants and for any r-admissible environment $\mathcal{Z}$ with input $1^{p'(\kappa)}$ $R_S^j(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) > 0$. So by Theorem 6 we have that the Fruitchain protocol is $(n/2-1,0,\delta)$-EVP under an r-admissible environment where $r \geq \frac{T_0}{p_f \cdot (\frac{n}{2}+1)\cdot(1-\delta)\cdot q}$. □

Note that:

- for any $\delta \in (0,1)$ and appropriate T_0 the Fruitchain protocol satisfies (T_0,δ)-approximate fairness property [108] [4].

- the above theorem holds not only if each fruit gives a fixed reward w_f but also when each fruit contributes to a reward pot an amount coming from transaction fees and flat reward and at the end the reward pot is shared evenly across miners of fruits. A similar setting is considered in [108] where the rewards of each single fruit are shared evenly among the miners of the fruits that belong to a preceding part of the chain that is long enough.

  However in [108] the total rewards are a fixed constant V (or almost V) in the whole execution. So our result is stronger in this respect. Additionally, regarding how our result compares to [108]: [108] includes the statement that *fairness* implies that in a sufficiently long window of the chain an adversary with relative hashing power $\rho < 1/2$ cannot have a fraction of blocks higher than $(1+\delta)\rho$ with overwhelming probability. In our case with the above theorem we prove formally that the Fruitchain protocol is EVP in the static synchronous setting for any coalition including fewer than half of the number of the participants and the approximation factor in the EVP is merely a constant additive factor (not multiplicative).

4.9.2 Absolute Rewards Minus Absolute Cost

We prove that when the cost c of each query is small enough compared to the reward w_f of each fruit then the Fruitchain protocol [108] is EVP for coalitions

[4] In [108] the number of queries q each participant can ask during each round is 1.

including even up to all but one participants and utility based on absolute rewards minus absolute cost.

Regarding how our work compares to the Fruitchain protocol [108] we note that in [108] the notion of *fairness* was used in order to prove incentive compatibility when the utility is based on absolute rewards (not absolute rewards minus absolute cost as we do) under the assumption that the total rewards are constant and the adversary controls fewer than half of the participants. By contrast, our result for absolute rewards minus absolute cost holds for any coalition including even up to all but one of the participants (this result can be easily transferred to utility equivalent to absolute rewards if we consider cost as zero). Moreover recall that in our result we consider that each fruit gives a fixed flat reward and we do not have the assumption that the total rewards are constant. It is an interesting research direction to explore what happens when we do not have the assumption of constant total rewards but we consider that each fruit contributes to a reward pot an amount coming from transaction fees and flat reward and at the end the reward pot is shared evenly across miners of fruits (a setting similar to [108]).

Theorem 8. *Assume that each fruit gives a constant reward and there exists* $\phi \in (0,1)$ *such that* $c < p_f \cdot w_f \cdot \phi$. *Then for any* $\delta_1 \in (0, 0.25)$, *such that* $c \leq p_f \cdot w_f \cdot (1-\delta_1) \cdot \phi$ *and* $4 \cdot \delta_1 < 1 - \phi$ *the Fruitchain protocol in a synchronous setting is* $(n-1, 4 \cdot \delta_1/(1-\phi), 0)$*-EVP according to the utility function absolute rewards minus absolute cost (cf. Def. 6) .*

Proof. Consider $\delta_1 \in (0, 0.25)$ such that $c \leq p_f \cdot w_f \cdot (1-\delta_1) \cdot \phi$ and $4 \cdot \delta_1 < 1 - \phi$. We choose also an arbitrary r-admissible environment $\mathcal{Z}$ with input $1^{p'(\kappa)}$, where κ is the security parameter and an arbitrary adversary $\mathcal{A}$ static with fixed cost that is PPT and it has corrupted a set T with t' participants, where $t' \in \{1, ..., n-1\}$. Note that if the adversary controls zero participants then the proof is trivial because adversary's utility is always zero. Let $x = \sum_{m=1}^{t'} x_m$ be the total number of the queries that all the corrupted participants collectively do not ask during each round. Note that x is a constant, not a random variable, as it is determined in the beginning by the static adversary. It holds $0 \leq x \leq q \cdot t'$.

We examine two executions of the Fruitchain protocol with the same environment, but with different adversary: in the first execution $\mathcal{E}_{\mathcal{Z},H_T}$ the adversary is H_T and in the second execution $\mathcal{E}'_{\mathcal{Z},\mathcal{A}}$ the adversary is $\mathcal{A}$. Note that the last complete round of the executions is r.

Firstly we have:

$$U_T^{min}(\mathcal{E}_{\mathcal{Z},H_T}) \geq q \cdot t' \cdot p_f \cdot r \cdot (1-\delta_1) \cdot w_f - c \cdot q \cdot t' \cdot r \geq q \cdot t' \cdot p_f \cdot r \cdot (1-\delta_1) \cdot w_f \cdot (1-\phi) > 0 \tag{4.43}$$

with overwhelming probability in κ.

The above equation is proved by Chernoff bound and taking into account that all the fruits produced by T is included in the local chain of all the honest participants at the end of the round r.

In addition, the adversary cannot earn rewards for more fruits than that it has produced. Moreover $c \leq p_f \cdot w_f \cdot (1-\delta_1) \cdot \phi$. As a result by Chernoff bound

$$U_T^{\max}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) \leq (q \cdot t' - x) \cdot p_f \cdot r \cdot (1+\delta_1) \cdot w_f - c \cdot (q \cdot t' - x) \cdot r \tag{4.44}$$

$$\leq q \cdot t' \cdot p_f \cdot r \cdot (1+\delta_1) \cdot w_f - c \cdot q \cdot t' \cdot r \tag{4.45}$$

with overwhelming probability in κ.

As a result

$$U_T^{\max}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) - U_T^{min}(\mathcal{E}_{\mathcal{Z},H_T}) \leq (\frac{1+\delta_1}{1-\delta_1} - 1) \cdot \frac{1}{1-\phi} \cdot U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \tag{4.46}$$

$$\leq 4 \cdot \delta_1 \cdot \frac{1}{1-\phi} \cdot U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \tag{4.47}$$

with overwhelming probability in κ. □

Note that:

- the lower c the better approximation (lower multiplicative approximation factor) we achieve.
- the expected number of blocks per round s that is connected to synchronicity does not affect the multiplicative factor as in our result in Bitcoin. This happens because we assume that there is no restriction in the number of the fruits included in a block so the adversary does not have an advantage if it sends its fruits first.
- recall that we have proved that also Bitcoin is EVP when the same utility is used and the difficulty of mining is fixed, something that does not happen when the difficulty changes after a period that depends on the number of blocks in the ledger. So it is an interesting research direction to prove that

the Fruitchain protocol is EVP even if the difficulty of mining changes after a period that depends on the number of blocks in the ledger .

- $\phi \in (0,1)$ such that $c < p_f \cdot w_f \cdot \phi$ reflects that the reward of each block is high enough compared to the cost of the mining.
- if $c = 0$ then $c < p_f \cdot w_f \cdot \phi$ holds for ϕ close to zero and the utility becomes absolute rewards (cf. Def. 6).

4.9.3 Overview of the Incentives in the Bitcoin and the Fruitchain Protocol

Summarizing Bitcoin is EVP for utility *absolute rewards* or *absolute rewards minus absolute cost*, coalitions up to $n-1$ participants where n is the number of the participants, fixed reward per block and fixed target or a target that changes after a fixed number of rounds that is independent of the number of the blocks that have been produced. On the other hand, Bitcoin is not an EVP with good approximation for these utilities, when the target of blocks changes after a specific number of blocks or the reward per block is not fixed due to transaction fees. Moreover Bitcoin is not an EVP for utility *relative rewards*.

The Fruitchain protocol [108] is EVP for utility *absolute rewards* or *absolute rewards minus absolute cost*, coalitions up to $n-1$ participants where n is the number of the participants and fixed reward per fruit. Note that the Fruitchain protocol considers fixed target of fruits and blocks. Moreover, the Fruitchain protocol is EVP for utility *relative rewards*, coalitions including fewer than $n/2$ and even if the reward per fruit is not fixed due to transaction fees.

Chapter 5

Reward Sharing Schemes for Stake Pools

5.1 Introduction

In the previous chapter we examined if some proof of work blockchain protocols are EVP. Note that in our analysis we took into account only the cost of mining (due to proof of work) and not the cost the miners/participants incur when they verify if the transactions they want to include in their blocks are valid. If we take into account the cost of verifying transactions, then the participants have incentives to create huge *mining pools* and share the cost of verification.

Mining pools in Bitcoin or other proof of work cryptocurencies work as follows: miners are organised into groups called mining pools and each group is controlled usually by a miner [1]. We will call this miner *pool leader*. All the miners/members of the pool: (i) try to extend the same block (ii) usually include in the block they try to produce the same set of transactions that is usually specified and verified by the pool leader (iii) try to produce blocks of lower difficulty than the current difficulty or in other words blocks of higher target (see Section 3.1.1.3 for the definition of target). These blocks are called *shares* and work as an indicator of the computational power they spend for mining for the pool. When a block of the current difficulty is produced by a member, then the rewards of this block are shared among the members of the pool by the pool leader depending on the number of the shares they have provided. In some cases the miners are paid per share they produce (see [112, 118] for more details regarding mining pools).

[1] or by a smart contract, see [94]

When we say that the miners share the cost of verification we mean that the pool leader verifies the transactions that will be included in the block the pool tries to mine and then when a block is produced, before the pool leader shares its rewards among the miners of the pool, it removes the cost it has incurred for this verification.

We leave as future work to prove that a joint strategy where all the miners mine in one pool is EVP both in Bitcoin and the Fruitchain protocol [108]. A number of previous works address this issue and critisize Bitcoin for its tendency to centralise via the creation of mining pools [11, 62, 86, 120].

This tendency to centralization appears also in proof of stake blockchain protocols. Towards this direction we prove in this chapter that if we use the reward mechanism of Bitcoin (in case all participants are honest) or the fair reward mechanism of the Fruitchain protocol in a setting that reflects a proof of stake protocol, then there is no Nash equilibrium with more than one pool.

Given that decentralization is one of the most important features of a blockchain protocol, in this chapter we design a reward mechanism that can be used in proof of stake blockchain protocols and satisfies two properties that are related to decentralization:

1. the first property is that in Nash equilibrium the participants form k pools of equal size, where k is a parameter.

2. the second property is related to resilience against Sybil attacks [43] or in other words to how many physical entities are the leaders of these pools. For example if in a Nash equilibrium there exist k pools but most of them are controlled by one physical identity, then our system cannot be considered decentralized.

Moreover, we would like our reward sharing mechanism to be as *cost efficient* as possible in the sense that the pools in the equilibrium belong to pool leaders with low cost. Note that in our results initially we use a proof of stake blockhain protocol as a "black box" abstracting its main characteristics and afterwards we discuss how our mechanism can be deployed in a proof of stake blockchain protocol. Moreover our setting is deterministic and the rewards are guaranteed to get shared among the participants as specified by the reward mechanism. For this reason we do not use our EVP model and we use other notion of Nash equilibrium.

5.1.1 Our Contribution and Roadmap

In this chapter:

1. we define a notion called *reward sharing scheme* (RSS) (Section 5.3).
2. we propose a reward sharing scheme that satisfies the two properties which we present above (Section 5.11).
3. we explain why other simpler reward schemes do not satisfy these properties (Section 5.4, 5.6).
4. we motivate our scheme (Section 5.5, 5.7, 5.9).
5. we define the game that we will use (Section 5.10) in our analysis and we prove formally our claims (Section 5.12).
6. we discuss deployment considerations of our reward sharing scheme in a proof of stake protocol (Section 5.13).
7. we present the trade off of our mechanism (Section 5.14).
8. we refer to related works (Section 6.1.2).

We note that this chapter is based on our paper [29] published in the IEEE European Symposium on Security and Privacy 2020 and the extended version of this paper in [28]. In [28, 29] we also experimentally demonstrate that a system that uses our reward sharing scheme will converge fast to the Nash equilibria we describe. These experiments were carried out by Lars Brünjes. More importantly the reward mechanism that has been deployed in tandem with the Ouroboros protocol [80] in the Incentivized Testnet and the Shelley update in Cardano launched by the company IOHK [3] (Input Output) is based on these results.

5.2 Our Setting

Assuming that we have a collaborative project such as maintaining a proof of stake blockchain, we consider a setting where:

- there are n stakeholders (aka participants or players). Each player owns some stake s_i in the project such that $\sum_{i=1}^{n} s_i = 1$. As the sum of players' stake is 1 with "stake" we mean relative stake.

- each player i can either participate in the project directly by creating a stakepool π_i and thus becoming *pool leader/pool operator* o and/or delegate its stake to another existing pool and thus becoming a *member* of this pool.

- when a player i creates a pool it incurs a cost c_i that is fixed and independent of the pool's size. This cost in a proof of stake blockchain protocol reflects the cost of running the server, verifying transactions, being online etc.

- the total stake of a pool π_i denoted by σ_i is the (relative) stake that has been delegated by other players to this pool and the stake the pool leader has allocated to this pool. Note that a pool π_i is created and is considered active only if the pool leader allocates non zero stake to it. Otherwise we will consider that it is inactive and thus $\sigma_i = 0$.

- the stake that player i delegates to a pool π_j created by player j is denoted by $a_{i,j}$.

- the higher σ_i is, the more frequently pool leader i participates in the project. Note that when we say that a player i delegates its stake to another pool we mean that it continues to own the stake but it delegates its "right for participation". For example consider a proof of stake blockchain protocol such as [80] where a pool is elected to sign a block with probability proportional to its total stake. In this case player i has the option to create a pool and get elected with probability s_i to sign a block or delegate some of its stake to another pool and in this way increase the probability with which the other pool is elected.

- the players are rational in the sense that they try to maximize their utility that in our case will be based on their profit.

- there is a fixed amount of rewards R that is shared among the players according to a *reward sharing scheme* (RSS) that we define in the next section. We assume that there are no other factors (*externalities*) that affect the rewards outside our reward scheme.

5.3 Definition of Reward Sharing Scheme (RSS)

A reward sharing scheme (RSS) consists of two levels:

1. the first level determines how the rewards R (or a subset of them) will be shared among the pools. To be more specific it includes a reward function that takes as input some characteristics of the pool and outputs its rewards. In the RSS that we propose these characteristics of a pool π_i are the total stake of the pool σ_i and the stake of the pool leader $\lambda_i = a_{i,i}$ called *pledged* stake. As we will explain later the *pledged stake* is needed as input in order to provide protection against Sybil attacks [43].

2. the second level determines how pool's rewards will be shared inside the pool. We examine reward sharing schemes where : (i) initially the pool leader is compensated for the cost it incurred by receiving an amount of the pool's rewards that is equal to this cost. In the case when the pool's rewards are not enough to compensate the pool leader for the cost it incurred, the remaining cost is paid by the pool leader. (ii) after that the remaining amount is shared among the members of the pool including the pool leader proportionally to the fraction of their stake in the pool.

Formally:

Reward sharing schemes (RSS) for stake pools: the class of reward sharing schemes we investigate is parameterised by a function $r : [0,1]^2 \to \mathbb{R}_{\geq 0}$ and operates as follows.

- The reward scheme distributes a total fixed amount R to the pools according to their stake σ_i and the stake of their pool leader $a_{i,i}$. In particular pool π_i gets reward $r(\sigma_i, a_{i,i})$ with $\sum_i r(\sigma_i, a_{i,i}) \leq R$. Note that we don't have to distribute the whole amount R. Formally, the function $r(\cdot,\cdot)$ takes the stake of a pool and the stake of the pool leader allocated to this pool and returns the payment for this pool so that: $\sum_i r(\sigma_i, a_{i,i}) \leq R$.

- $r(0,0) = 0$, which means that a pool with no stake will get zero rewards.

- The reward $r(\sigma_i, a_{i,i})$ of each pool π_i is shared among its pool leader and its stakeholders. This may be done in a number of ways but in any case, the pool leader should get an amount at least $c_i^- = \min(c_i, r(\sigma_i, a_{i,i}))$ to cover the declared cost for running the pool. We will focus our investigation on reward schemes that are *proportional*, i.e., those schemes that have the property that the ratio of the rewards obtained by stakeholder j_1 over the

rewards of stakeholder j_2 in pool π_i equals $a_{j_1,i}/a_{j_2,i}$, with the only exception being for pool leaders who may be considered for additional rewards.

Definition 10. *A pool will be called saturated if it has total stake at least* $1/k$*, where* k *is the parameter that is related to the desired number of pools.*

5.4 Unsuitable Reward Functions for k Pools in a Nash Equilibrium

5.4.1 Summary of our Results

Recall that our target is to design a reward sharing scheme (i) such that when the utility of the players is the profit, in Nash equilibrium there will exist k pools of equal size and (ii) that is *Sybil resilient*.

Searching for the suitable reward function for the first level of our RSS, we have excluded some classes of reward functions because we have proved that they lead to Nash equilibria with at most one pool or a number of pools equal to the number of players.

The class of functions that we excluded are related to the quantity $\frac{r(\sigma,\lambda)-c}{\sigma}$. The motivation for selecting to examine this quantity is the following: when a pool π_i earns rewards $r(\sigma_i, a_{i,i})$ that are higher than the cost of the pool c_i then each player that has allocated stake s to this pool will earn an amount of rewards that is equal to $s \cdot \frac{r(\sigma_i,a_{i,i})-c_i}{\sigma_i}$.

In a nutshell, we prove that:

- when $\frac{r(\sigma,\lambda)-c}{\sigma}$ is strictly increasing in the whole domain of σ, then there is no equilibrium with more than one pools.

  Note that this class is important because it includes the *fair* reward function where each pool gets rewarded proportionally to its stake. This means that if we use in our setting a reward function that reflects the reward mechanism of Bitcoin when all players play honestly and the reward mechanism of the Fruitchain protocol [108] then there will exist no Nash equilibrium with more than one pools.

- when $\frac{r(\sigma,\lambda)-c}{\sigma}$ is strictly decreasing in the whole domain of σ then there are instances of the stake and cost distributions so that there is no Nash equilibrium with a number of pools smaller than the number of players.

At this point we present (i) a description of the game and the utilities function that we use in order to prove the above results (ii) the theorems and the proofs that verify the above result. Initially for simplicity we prove the negative result for the fair reward function and after that we present the more general result.

5.4.2 The Stake Pools Game and the Utility Function

We define the stake pools game where the strategies of the players are their allocations of their stake to their own as well as the other available pools. In this game each player i tries to maximize its utility. Recall that the rewards of a pool π_i are $r(\sigma_i, a_{i,i})$ and the cost the pool leader/operator incurs for running this pool is c_i. The pool operator gets its cost reimbursed, apart from that, all rewards are split proportional to stake. So if a player i with cost c_i runs a pool with total stake σ_i, its utility $u_{i,i}$ from this pool π_i is

$$u_{i,i} = \begin{cases} r(\sigma_i, a_{i,i}) - c_i & \text{for } r(\sigma_i, a_{i,i}) \leq c_i, \\ \frac{a_{i,i}}{\sigma_i} \cdot (r(\sigma_i, a_{i,i}) - c_i) & \text{otherwise,} \end{cases}$$

and a player $j \neq i$ delegating stake $a_{j,i}$ to that pool π_i gets rewards

$$u_{j,i} = \begin{cases} 0 & \text{for } r(\sigma_i, a_{i,i}) \leq c_i, \\ \frac{a_{j,i}}{\sigma_i} \cdot (r(\sigma_i, a_{i,i}) - c_i) & \text{otherwise} \end{cases}$$

from that pool. We define the utility of each player j to be $u_j = \sum_i u_{j,i}$.

5.4.3 Fair RSS's and their Failure to Decentralise

We show that if we use a "fair" reward sharing scheme, then we end up in Nash equilibrium with at most one pool, which means that this scheme fails our decentralization objective.

Specifically consider the fair allocation that sets $r(\sigma_i, a_{i,i}) = \sigma_i \cdot R$, i.e., pools are rewarded proportionally to their size. For simplicity we take $R = 1$. (Note that if we consider $R = 1$ then all the costs are between zero and one.) Moreover, we assume that all pool participants are also treated fairly receiving rewards proportionally to the stake they have delegated in the pool of their choice.

We prove the following theorem:

Theorem 9. *Given the above reward sharing scheme assuming that there exists at most one player with zero cost we have the following:*

(I) there is no equilibrium where more than one pool is created.

(II) if there exists i such that $s_i > c_i$ then the only Nash equilibria are the following: there exists just one pool, say π_i and it holds (i) $c_i \leq 1$ and (ii) $s_j \cdot c_i \leq c_j$ for each member j of this pool (iii) all players have delegated their stake to π_i.

Note that in (II) we use the assumption that there exists i such that $s_i > c_i$ in order to exclude equilibria with zero pools.

Proof. (I) We prove it by contradiction. Let us suppose that there is a Nash equilibrium where there exist l pools with $l > 1$. We order the pools according to the quantity $\frac{c_i}{\sigma_i}$. Let π_{j_0} be the pool with the lowest value of this quantity denoted by $\frac{c_{j_0}}{\sigma_{j_0}}$. Then the members of all the other pools have incentives to delegate their stake to π_{j_0}. So this cannot be an equilibrium. Specifically let us examine a pool member j of a pool π_i. It holds $\frac{c_i}{\sigma_i} \geq \frac{c_{j_0}}{\sigma_{j_0}}$. The utility of player j if they leave their stake $a_{j,i}$ in π_i is $\frac{a_{j,i}}{\sigma_i} \cdot (\sigma_i - c_i) = a_{j,i} - a_{j,i} \cdot \frac{c_i}{\sigma_i}$. Note that $\sigma_i - c_i$ is non negative, because otherwise this would not be an equilibrium, as the pool leader could increase its utility by dissolving its pool and leaving its stake unallocated. If they remove their stake from π_i and delegate it to π_{j_0} their utility is $\frac{a_{j,i}}{\sigma_{j_0}+a_{j,i}} \cdot (\sigma_{j_0} + a_{j,i} - c_{j_0}) = a_{j,i} - a_{j,i} \cdot \frac{c_{j_0}}{\sigma_{j_0}+a_{j,i}}$ which is strictly higher, because $\frac{c_i}{\sigma_i} \geq \frac{c_{j_0}}{\sigma_{j_0}}$. Note that we have assumed that there exists at most one player with zero cost. So if there exists a player with zero cost then this player is j_0.

(II) Firstly we prove that the joint strategies that satisfy properties (i),(ii) and (iii) as defined by the theorem are indeed Nash equilibria.

The pool leader has no incentives to dissolve its pool because its utility is $s_i - s_i \cdot \frac{c_i}{\sigma_i}$ that is greater or equal to zero, because $1 \geq c_i$ and $\sigma_i = 1$.

The pool members have no incentives to create their own pool because their current utility is $s_j \cdot (1 - c_i) = s_j - s_j \cdot \frac{c_i}{1}$ and is not lower than the utility they get if they create their own pool which is equal to $s_j - c_j$, given that $s_j \cdot c_i \leq c_j$ and $\sigma_i = 1$.

Secondly we prove that there is no other Nash equilibrium. By Theorem 9 I we have that there is no equilibrium with more than one pools. We prove by contradiction that (a) there is no equilibrium with zero pools and (b) there is no equilibrium with one pool but without (i),(ii) and (iii) properties:

(a) Let us assume that there is an equilibrium with no pools. Then player i_0 for whom it holds $s_{i_0} > c_{i_0}$ can increase its utility by creating its own pool. (b) Let us assume that there is an equilibrium with one pool where (i) or (ii) does

not hold. If (i) does not hold, then the pool leader of the pool has negative utility and thus they have incentives to dissolve their pool. If (ii) does not hold then a pool member can increase its utility by creating its own pool.

Now we prove that an arbitrary equilibrium with one pool satisfies (iii). We consider that in an equilibrium player i_0 for whom it holds $s_{i_0} > c_{i_0}$ is a pool member or a pool leader of the unique pool π_i. This holds because if they did not participate at all then they could increase their utility by creating their own pool, but this cannot happen as we are in an equilibrium. As this player is pool leader or pool member the profit of the pool π_i $\sigma_i - c_i$ is positive (this holds because otherwise player i_0 has incentives to create its own pool). So other players cannot have chosen to not participate at all in the equilibrium (which means that (iii) holds) because otherwise they could increase their utility by delegating to π_i. □

5.4.4 Reward Functions That Lead to Too Many Pools or Just One Pool

Theorem 10. *I) If* $\frac{r(\sigma,\lambda)-c}{\sigma}$ *as a function of* σ *is strictly increasing in* $\sigma \in (0,1]$ *then there is no equilibrium with more than one pool. Note that a fair reward function* $r(\sigma,\lambda) = \sigma$ *is such an example.*

II) If $r(\sigma,\lambda) = r(\sigma)$ *a continuous and strictly increasing function on* σ *and* $\frac{r(\sigma,\lambda)-c}{\sigma}$ *as a function of* σ *is strictly decreasing in* $\sigma \in [\sigma_0,1]$*, for* σ_0 *such that* $r(\sigma_0)-c>0$*, then there is an assignment of costs and stakes to the players such that there is no equilibrium with fewer than* n *pools where* n *is the number of players. We assume for the proof that each player can delegate to a pool stake at least* $\frac{s_{min}}{f}$ *where* $f \in (1,\infty)$ *and* s_{min} *is the minimum stake among all the players.*

Proof. I) We prove it by contradiction. Let us assume that there is an equilibrium where there are $l > 1$ pools. Then we order the pools according to the quantity $\frac{r(\sigma_i,a_{i,i})-c_i}{\sigma_i}$. Let π_{j_0} be the pool with the highest value. Then this cannot be an equilibrium given that the members of the other pools have incentives to delegate their stake to π_{j_0}. In more detail, the utility of a player j that leaves its stake to pool π_i is $\frac{r(\sigma_i,a_{i,i})-c_i}{\sigma_i}\cdot a_{j,i}$ and if they remove it and reallocate it to pool π_{j_0} then it becomes $\frac{r(\sigma_{j_0}+a_{j,i},a_{j_0,j_0})-c_{j_0}}{\sigma_{j_0}+a_{j,i}}\cdot a_{j,i}$ that is strictly higher.

II) Consider $f \in (1,\infty)$. We find an assignment of costs and stake to the players so that there is no equilibrium with a number of pools smaller than the number

of players n. In more detail: (i) the assignment of stake is such that each player has stake $\frac{1}{n}$ (ii) the assignment of costs is arbitrary except the minimum cost c_{min} and the maximum cost c_{max} that we determine. Specifically we take c_{max} such that $max\{r(\frac{s_{min}}{f}) - \frac{1}{f} \cdot r(s_{min}), 0\} < c_{max} < r(\frac{s_{min}}{f})$ and we determine c_{min} as follows:

- It holds $\frac{r(\frac{s_{min}}{f}) - c_{max}}{\frac{s_{min}}{f}} > \frac{r(\frac{1}{n}) - c_{max}}{\frac{1}{n}}$ because $r(\frac{s_{min}}{f}) > c_{max}$, $\frac{s_{min}}{f} < \frac{1}{n}$ and $\frac{r(\sigma, \lambda) - c}{\sigma}$ as a function of σ is strictly decreasing.
- Consider r_0 such that $y = \frac{r(\frac{s_{min}}{f}) - c_{max}}{\frac{s_{min}}{f}} > r_0 > \frac{r(\frac{1}{n}) - c_{max}}{\frac{1}{n}}$.
- We define the function $g(x) = \frac{r(\frac{1}{n}) - x}{\frac{1}{n}}$ which is continuous in $[c, c_{max}]$, for c such that $\frac{r(\frac{s_{min}}{f}) - c_{max}}{\frac{s_{min}}{f}} = \frac{r(\frac{1}{n}) - c}{\frac{1}{n}}$.
- We know that there exists $c \in (0, c_{max})$ such that $y = g(c)$ by the *intermediate value theorem* because (i) $g(c_{max}) < y < g(0)$ given that $max\{r(\frac{s_{min}}{f}) - \frac{1}{f} \cdot r(s_{min}), 0\} < c_{max} < r(\frac{s_{min}}{f})$ and (ii) $g(x)$ continuous in $[0, c_{max}]$. Note that $s_{min} = \frac{1}{n}$ given that all players have stake $\frac{1}{n}$.
- By the *intermediate value theorem* again there is $x \in (c, c_{max})$ such that $g(c) > g(x) = r_0 > g(c_{max})$.
- We set $c_{min} = x$ the minimum cost among all players. Note that it holds $g(c) = \frac{r(\frac{s_{min}}{f}) - c_{max}}{\frac{s_{min}}{f}} > \frac{r(\frac{1}{n}) - c_{min}}{\frac{1}{n}} = g(c_{min})$.

Now we prove that for these values of c_{min}, c_{max} and assignment of stake ($1/n$ for each player) there is no equilibrium with fewer than n pools.

- Firstly we prove $C \Rightarrow A \cup B$ where C is the event where there exist fewer than n pools, A the event where there exists at least one player that has left some of its stake unallocated and B the event where there exists at least one pool with stake more than $1/n$. In order to prove $C \Rightarrow A \cup B$ we prove $\neg(A \cup B) \Rightarrow \neg C$ which is equivalent to $\neg A \cap \neg B \Rightarrow \neg C$. Let us assume that there is no stake unallocated (so the total stake that is delegated to pools is 1) and all pools have stake at most $1/n$. If x is the number of the pools then the total stake that is delegated is at most $x \cdot \frac{1}{n}$ which means that x is at least n.
- Now in order to prove that there is no equilibrium with fewer than n pools it is sufficient to prove that there is no equilibrium where the event A or the event B happens.

- Firstly we prove by contradiction that there is no equilibrium where the event B happens. Let us assume that there exists such an equilibrium where a pool π_i has stake more than $1/n$. Then this pool has pool members given that $s_i = \frac{1}{n}$. Then player j who has delegated to this pool stake $a_{j,i} \geq \frac{s_{min}}{f}$ has incentives to remove stake $\frac{s_{min}}{f}$ from pool π_i and create its own pool. This happens because the utility of player j from pool π_i for this part of its stake is smaller than $\frac{r(\frac{1}{n})-c_{min}}{\frac{1}{n}} \cdot \frac{s_{min}}{f}$ which is smaller than $\frac{r(\frac{s_{min}}{f})-c_{max}}{\frac{s_{min}}{f}} \cdot \frac{s_{min}}{f}$ and thus smaller than its utility if they create its own pool with stake $\frac{s_{min}}{f}$.

- We can easily prove by contradiction that there is no equilibrium where the event A happens. Let us assume that there is a player j that has left some of its stake unallocated. If this part of stake is $\frac{s_{min}}{f}$ or more then they can increase their utility by creating their own pool given that $r(\frac{s_{min}}{f}) - c_{max} > 0$. If this player has some stake x smaller than $\frac{s_{min}}{f}$ unallocated then this means that either (i) they have allocated stake to another pool and thus there exists a pool with stake more than $1/n$ (this leads to contradiction as we have proved above) or (ii) they have created their own pool. In the latter case this player has incentives to include the unallocated part of stake to this pool given that the function r is strictly increasing ($r(s_j) > r(s_j - x)$). Note that this pool does not have other members as we have proved above.

□

5.5 Motivation for Cap in our Reward Function

Based on the above results we conclude that we should choose a reward function that makes $\frac{r(\sigma,\lambda)-c}{\sigma}$ strictly increasing up to a point that we will call *cap* or *saturation point* and strictly decreasing afterwards. This point will be $1/k$ that is the size a pool has in a Nash equilibrium when there exist k pools of equal size. The intuition behind the cap is that we incentivize pools to grow until their size becomes $1/k$ and we disincentivize them to grow further. Thus the reward function that we choose is increasing up to the cap and constant afterwards.

5.6 Unsuitable Reward Functions for Sybil Resilience

Based on the above, a simple reward function with cap is $r(\sigma, \lambda) = \min\{\sigma, 1/k\}$. This reward function seems to lead to a Nash equilibrium with k pools of equal size that belong to the players with the lowest cost.

However, observe that it does not take into account at all the stake of the pool leader. This means that if we allowed users/agents (i) to create multiple identities pretending multiple players (Sybil attack [43]) (ii) to declare a fake cost for these identities, then most of the pools in the Nash equilibrium could belong to the same user.

Thus the reward function we propose takes into account also the stake of the pool leader. How much influence the stake of the pool leader has on the pool's rewards depends on a *Sybil resilience parameter* α that imposes a tradeoff between efficiency and Sybil resilience. The higher this parameter is, the better Sybil resilience we provide but the less efficient our system will be. With "efficient" we mean the cost of the pool leaders that run the pools in a Nash equilibrium.

5.7 Motivation for Margin

Observe that a player who chooses to create its own pool bears an extra risk compared to players who are members. If the pool does not get enough rewards to cover its cost then this loss will be paid by the pool leader. Thus we should allow the pool leaders to ask for an extra fraction of the pool's profit compared to the other members in order to get compensated for their risk. This fraction will be called *margin* and will be chosen strategically from the pool leader. Note that the pool leader should choose a suitable value for the margin that makes its pool competitive.

5.8 Reward Sharing Scheme with Cap and Margin

A reward scheme for stake pools that incorporates the above features will be called *reward sharing scheme with cap and margin*. Formally:

Definition 11 (Reward sharing schemes with cap and margin)**.** *A reward sharing scheme with cap and margin is a reward sharing scheme that (1) is parameterised by a function* $r : [0,1]^2 \to \mathbb{R}_{\geq 0}$ *(that takes as input the stake* σ_i *of a pool* π_i *and*

the stake $a_{i,i}$ of the pool leader allocated to this pool and returns the total reward for this pool) and a value $k \in \mathbb{N}$ and satisfies the following properties:

- *(as before)* $\sum_{i=1}^{n} r(\sigma_i, a_{i,i}) \leq R$*, where* R *are the total rewards.*
- *(as before)* $r(0,0) = 0$*.*
- $\frac{d[(r(\sigma,\lambda)-c)\cdot\frac{1}{\sigma}]}{d\sigma} > 0$*, when* $\sigma \leq \beta \stackrel{\text{def}}{=} \frac{1}{k}$*. This means that the reward function is increasing for small values of pool's stake to incentivize players to join together in pools to share the cost.*
- $\forall\lambda\ r(\sigma,\lambda) = r(\beta,\lambda)$ *when* $\sigma > \beta$*. This means that the reward function is constant for large values of the pool's stake to discourage the creation of large pools.*

(2) the reward $r(\sigma_i, a_{i,i})$ of each pool π_i is shared among its pool leader and its stakeholders. The pool leader gets an amount $c_i^- = \min(c_i, r(\sigma_i, a_{i,i}))$ to cover the declared cost for running the pool. A fraction m_i of the remaining amount $(r(\sigma_i, a_{i,i}) - c_i^-)$ is the pool leader compensation for running the pool. This fraction is referred to as margin. The rest $(1 - m_i) \cdot (r(\sigma_i, a_{i,i}) - c_i^-)$ is distributed to the stakeholders of the pool, including the pool leader, proportionally to their contributed stake.

5.9 Motivation for Non-Myopic Utility

Observe that the addition of margin makes the game much more complicated because the pool leaders have an extra strategy which is to select their margin. If we consider in this game as utility the profit, as in the previous game, then we have the following implications:

1. there is no Nash equilibrium with margin lower than 1; if all but one players cannot change their strategy, then the remaining player can increase its utility by making its margin 1 given that no member will leave the pool.

   However this does not seem a stable state: If all margins are 1, a non-myopic player (a forward-looking player who tries to predict the final size of the pools after the other players play) who is not a pool leader can create a new pool with smaller margin which will attract enough stake to make it profitable.

2. apart from the margin that should be selected non-myopicly by the pool leaders, the delegators should also think in a non-myopic way, because otherwise it is difficult for new pools to "enter the market" and attract delegators, something that could lead to large pools with "bad characteristics" such as high margin, high cost, low pledged stake.

   In more detail, when some saturated pools (pools with stake at least $1/k$) have already been created, then it is very difficult for a new pool to become competitive in terms of rewards and attract delegators, even if it has "better characteristics" than the existing pools (with "better characteristics" we mean lower margin, lower cost and higher pledged stake; when a pool π_i has better characteristics than another pool, then when these two pools have the same size, pool π_i gives more rewards to its members). The reason is that the size of the new pool is very small, because initially includes only the stake of the pool leader (recall that for pools smaller than the cap $1/k$ it holds that the higher total stake they have, the more rewards the pool earns). As a result players who want to delegate their stake may prefer the already existing saturated pools.

   Observe that in such a scenario the pool leader could increase its margin in order to avoid oversaturation but:

   (a) the existing pool leaders may want to prevent new pools to "enter the market" in order to avoid competition in the future.

   (b) a malicious pool leader may prefer an oversaturated pool in order to get elected more often and can attack the system.

For these reasons, in order to analyse our *reward sharing schemes with cap and margin* we will use a natural *non-myopic* type of utility which enables the players to be more far-sighted. Specifically, *a player computes its utility using the estimated final size of the pools instead of the current size of the pools.* A non-myopic player thinks in the following way: if all the pools had the same size $1/k$ (which is size of pools we want in the Nash equilibrium) which are the k pools that would give the highest rewards to their members? This quantity will be called *desirability.* The prediction of a non-myopic player would be that (i) the k pools with the highest desirability will become saturated (if they are not already) which means that their estimated size is the *maximum between their*

current size and $1/k$ (ii) the estimated size of the other pools is just their pledged stake with the stake of a potential delegator.

At this point we will describe formally the extended stake pools game that includes margin, as well as the strategies and the non-myopic utility of the players.

5.10 The Stake Pools Game with Cap and Margin

Without loss of generality we assume that (i) every player can be the leader of only one pool and (ii) each player has stake at most $\beta = 1/k$. Players that have stake more than β or wish to create more than one pool can be thought as a strategic coalition of players which we analyse in the Section 5.11 where we consider Sybil attacks of this nature.

We use the notation: $(x)^+ = \max(0,x)$, and $[n] = \{1,\ldots,n\}$.

Definition 12 (Strategy of a player)**.** *The strategy of a player* i *has two parts:*

- (m_i, λ_i)*, where* $m_i \in [0,1]$ *is the margin and* λ_i *the stake that player* i *will commit if it activates its own pool.*
- $S_i^{(\vec{m},\vec{\lambda})} = \vec{a}_i^{(\vec{m},\vec{\lambda})}$ *that is the allocation of player* i*' stake given* $(\vec{m},\vec{\lambda})$*. When the* $(\vec{m},\vec{\lambda})$ *can be inferred from the context we will use* $\vec{a}_i$ *for simplicity.* $a_{i,j} \in [0,1]$ *denotes the stake that player* i *allocates to pool* π_j *so that its total allocated stake is* $\sum_{j=1}^{n} a_{i,j} \leq s_i$*. This allows for stake* $s_i - \sum_{j=1}^{n} a_{i,j}$ *of the player to remain unallocated. In addition* $a_{i,i}^{(\vec{m},\vec{\lambda})} \in \{0, \lambda_i\}$.

Definition 13 (Pools)**.** *Given a joint strategy* $\vec{S}^{(\vec{m},\vec{\lambda})}$*, the stake allocated to a pool* π_j *is denoted by* $\sigma_j(\vec{S}^{(\vec{m},\vec{\lambda})})$*, or simply* σ_j *for a less cluttered notation. A pool* π_j *is called* active *when player* j *allocates non-zero stake to it, that is,* $a_{j,j} = \lambda_j > 0$*. Note that only player* j *can activate pool* π_j*. If a pool* π_j *is active its stake is* $\sigma_j = \sum_{i=1}^{n} a_{i,j}$*, otherwise we assume that* $\sigma_j = 0$*. A pool is called* saturated *when its stake is at least* β .

The restriction that only player j can activate pool π_j, by allocating non-zero stake to it, is necessary to prevent other players to force player j pay the cost c_j of operating the pool without consenting to open the pool.

Definition 14 (Desirability and Potential Profit)**.** *The potential profit of a saturated pool with allocated pool leader stake* λ *and cost* c *is* $P(\lambda, c) = r(\beta, \lambda) - c$.

Given a joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$, we define the desirability of a pool π_j

$$D_j(\vec{S}^{(\vec{m},\vec{\lambda})}) = \begin{cases} (1-m_j)P(\lambda_j, c_j) & \text{if } P(\lambda_j, c_j) \geq 0 \\ 0 & \text{elsewhere} \end{cases} \tag{5.1}$$

Note that the desirability of a pool depends on its margin, the stake of the pool leader allocated to this pool and its cost.

Definition 15 (Ranking)**.** *Given a joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$, the rank of a pool π_j denoted by $\text{rank}_j(\vec{S}^{(\vec{m},\vec{\lambda})})$ is its ranking with respect to the desirability $D_j(\vec{S}^{(\vec{m},\vec{\lambda})})$. The maximum desirability gets rank 1, the second maximum desirability gets rank 2, etc. Again to get a less cluttered notation, we will write rank_j instead of $\text{rank}_j(\vec{S}^{(\vec{m},\vec{\lambda})})$ whenever the joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$ can be inferred from the context. Ties break according to the potential profit, specifically the pool with the higher potential profit will be ranked higher; (with higher we mean smaller rank) for convenience we assume that all potential profit values are distinct. The k most desirable pools will be these ones with rank smaller or equal to k.*

Given the ranking, we define the non-myopic stake of a pool to be either the stake allocated by the pool leader or the size of a saturated pool. The first one is used when the pool does not belong to the k most desirable pools and the second one when the pool is among them.

Definition 16 (Non-myopic stake)**.** *The non-myopic stake of pool π_j is defined as*

$$\sigma_j^{\text{NM}}(\vec{S}^{(\vec{m},\vec{\lambda})}) = \begin{cases} \max(\beta, \sigma_j) & \text{if } \text{rank}_j \leq k \\ a_{j,j} & \text{otherwise.} \end{cases} \tag{5.2}$$

To simplify the notation we use σ_j^{NM} instead of $\sigma_j^{\text{NM}}(\vec{S}^{(\vec{m},\vec{\lambda})})$, σ_j instead of $\sigma_j(\vec{S}^{(\vec{m},\vec{\lambda})})$, rank_j instead of $\text{rank}_j(\vec{S}^{(\vec{m},\vec{\lambda})})$ and $a_{j,j}$ instead of $a_{j,j}(\vec{S}^{(\vec{m},\vec{\lambda})})$.

Definition 17 (Non-myopic utility)**.** *The utility $u_i(\vec{S}^{(\vec{m},\vec{\lambda})})$ of player i from being a member of pool π_j with non-myopic stake σ_j^{NM} is*

$u_{i,j}(\vec{S}^{(\vec{m},\vec{\lambda})}) =$

$$\begin{cases} 0 & \pi_j \text{ is inactive } (a_{j,j} = 0) \\ (1-m_j)\left(r(\beta, \lambda_j) - c_j\right)^+ \frac{a_{i,j}}{\sigma_j^{\text{NM}}} & \text{rank}_j \leq k \wedge a_{j,j} \neq 0 \\ (1-m_j)\left(r(\lambda_j + a_{i,j}, \lambda_j) - c_j\right)^+ \frac{a_{i,j}}{\lambda_j + a_{i,j}} & \text{otherwise.} \end{cases}$$

The utility $u_j(\vec{S}^{(\vec{m},\vec{\lambda})})$ *that the pool leader* j *gets from pool* π_j *is* $u_{j,j}(\vec{S}^{(\vec{m},\vec{\lambda})}) =$

$$\begin{cases} 0 & \pi_j \text{ is inactive } (a_{j,j}=0) \\ r(\sigma_j^{\mathrm{NM}},\lambda_j)-c_j & r(\sigma_j^{\mathrm{NM}},\lambda_j)-c_j<0 \wedge a_{j,j}\neq 0 \\ (r(\sigma_j^{\mathrm{NM}},\lambda_j)-c_j)\left(m_j+(1-m_j)\frac{\lambda_j}{\sigma_j^{\mathrm{NM}}}\right) & \text{otherwise.} \end{cases}$$

The utility of player i *is the sum of the utilities coming from all pools in which it participates as a pool leader or a pool member:* $u_i(\vec{S}^{(\vec{m},\vec{\lambda})}) = \sum_{j=1}^{n} u_{i,j}(\vec{S}^{(\vec{m},\vec{\lambda})})$.

Note that when a pool has rank higher than k, then the stake used for computing the non-myopic utility of a member who wishes to delegate s stake to this pool is the pledged stake plus s.

5.11 A Sybil Resilient Reward Sharing Scheme

5.11.1 Sybil Behavior and Resilience

In this subsection we present the second property that we want our RSS to satisfy that is related to Sybil resilence and how many physical entities are the pool leaders of the k pools in a Nash equilibrium.

Our goal is to disincentivize Sybil behaviour [43] where an agent creates multiple identities and declares potentially lower costs for each one. We distinguish two types of Sybil behaviors:

1. the first one captures a utility non-maximizer whose goal is to control 50% of the system. Note that in a proof of stake blockchain protocol where the possibility with which the pool leader of a pool is elected to sign a block is proportional to the total stake of its pool, a utility non-maximizer can control 50% of the system by creating pools that all together have attracted stake more than 1/2. Such level of control enables a player to invalidate the security properties of the blockchain protocol (see [80]).

2. the second type of Sybil behavior is that of a utility maximizer who creates multiple identities and their corresponding pools share the same server back-end and thus also the operational costs. Such a player reduces the number of independent server deployments that provide the service and

thus limits decentralisation. Note that this also can include coalitions of players that act as one.

This is a behavior that we cannot exclude in the anonymous setting that we operate. The best possible that we can hope for is to lower bound the stake of the Sybil player to be *linear* in the number of identities that it creates. So the second property that we want our RSS to satisfy is that it allows us to provide a *linear* in the number of identities lower bound of the stake $s_{\min}$ needed for the Sybil behavior to be effective.

Recall that in order for our reward sharing scheme to satisfy this property, our reward function takes as input also the pledged stake of the pool leader. This guarantees that players can attract stake from other players only if they commit substantial stake to their own pool.

The simplest reward function that takes into account the pledged stake of the pool leader would be $r(\sigma,\lambda) \sim \min\{\sigma,\beta\} + \alpha\lambda$, where α is a nonnegative parameter.

However this reward function has the downside that the influence of the stake λ of the pool leader when a pool is still small is very significant.

So the reward function that we choose (i) has the characteristic that the more total stake a pool has, the more impact pledged stake λ has on the pool's rewards and (ii) when pools have the ideal size $1/k$, the pledged stake is multiplied by just α, which is quite desirable when a parameterisation is attempted and the value of α will be used to control Sybil attacks.

5.11.2 Our RSS Construction

Given our target number of pools k, we define the reward function $r_k : [0,1]^2 \to \mathbb{R}_{\geq 0}$ of a pool π with stake σ and pool leader's allocated stake λ as follows:

$$r_k(\sigma,\lambda) = \frac{R}{1+\alpha} \cdot [\sigma' + \lambda' \cdot \alpha \cdot \frac{\sigma' - \lambda' \cdot (1-\sigma'/\beta)}{\beta}],$$

where $\lambda' = \min\{\lambda,\beta\}$, $\sigma' = \min\{\sigma,\beta\}$ and β,α are fixed parameters. A natural choice is $\beta = 1/k$, where k is the target of number of pools. For simplicity we will write r instead of r_k.

We have: $\alpha \in [0,\infty)$, $k \in \mathbb{N}, (k < n)$ and $R \in \mathbb{R}$. Note that the total rewards R and α should be selected such as it holds also $P(s_{k+1}, c_{k+1}) > 0$.

The next proposition shows that the proposed function is suitable for a *reward sharing scheme with cap and margin*.

Proposition 1. *The function $r(\cdot,\cdot)$ satisfies the properties of a reward sharing scheme with cap and margin (cf. Def. 11).*

Proof. It holds $\forall i\ r(\sigma_i, a_{i,i}) \geq 0$, as $a'_{i,i} \leq \sigma'_i$ and also:

1. $\sum_{i=1}^{n} r(\sigma_i, a_{i,i}) \leq R$, as $\frac{\sigma'_i - a'_{i,i} \cdot \frac{(\beta - \sigma')}{\beta}}{\beta} \leq 1$ and $\sum_{i=1}^{n}[\sigma_i + a_{i,i} \cdot \alpha] = \sum_{i=1}^{n} \sigma_i + \alpha \cdot \sum_{i=1}^{n} a_{i,i} \leq 1 + \alpha$.
2. $r(0,0) = 0$.
3. When $\sigma \leq \beta$ it holds: $\frac{d[r(\sigma,\lambda) - c) \cdot \frac{1}{\sigma}]}{d\sigma} > 0$.
4. $\forall \lambda\ r(\sigma,\lambda) = r(\beta,\lambda)$, when $\sigma > \beta$ because we have $\sigma' = \min\{\sigma,\beta\}$.

This completes the proof. □

5.12 Proof That our RSS Satisfies the Two Desired Properties

In this section we prove that our RSS satisfies the two desired properties. Initially we prove that forming k pools of equal size is a Nash equilibrium and after that we prove that our RSS is Sybil resilient as we can provide the desired lower bound. In the following section we also justify that there is a tradeoff between Sybil resilience and cost efficiency.

In our analysis we consider that the players are ordered in terms of potential profit, e.g., player 1 is the player with the highest $P(s_i, c_i)$.

5.12.1 Nash Equilibria of the Stake Pools Game

In this subsection we prove that forming k pools of equal size is a Nash equilibrium.

5.12.1.1 Perfect Strategies

We define a class of strategies and we prove that they are *Nash equilibria* of our game (Theorem 11). This class has the following characteristics: exactly k pools

of equal size are created and the pool leaders are the players with the highest value $P(s,c)$ (when $\alpha = 0$ those are the players with the smallest cost). Recall also that players decide to create or not a pool and how much stake they will allocate to other pools. In addition they decide a margin for their potential pool.

Definition 18 (Perfect strategies)**.** *We define a class of strategies, which we will call perfect. The margins are*

$$m_j^* = \begin{cases} 1 - \frac{P(s_{k+1},c_{k+1})}{P(s_j,c_j)} & \text{when } j \leq k \\ 0 & \text{otherwise,} \end{cases}$$

the stake allocated by each pool leader to their own pool is equal to their whole stake and the allocations are such that each of the first k pools has stake β.

Note that when $j \leq k$ it holds $\text{rank}_j \leq k$.

The following proposition gives the utilities at perfect strategies and it follows directly from Definition 17 of the non-myopic utilities of pool members and pool leaders and our reward function described in this section.

Proposition 2. *In every perfect strategy, (i) the utilities of the players are:*

$$u_i = P(s_{k+1},c_{k+1})\frac{s_i}{\beta} + (P(s_i,c_i) - P(s_{k+1},c_{k+1}))^+, \tag{5.3}$$

and (ii) the desirability of the first $k+1$ players is the same *and equal to $P(s_{k+1},c_{k+1})$.*

To justify the proposition note that all the players get a fair reward, in the sense that it is a constant $P(s_{k+1},c_{k+1})/\beta$ times their stake, with the exception of each pool leader i, who gets an additional reward $P(s_i,c_i) - P(s_{k+1},c_{k+1})$. This additional reward can be viewed as a bonus for the efficiency and security that the pool leader brings to the system. We will show that every perfect strategy is a Nash equilibrium of the game with the defined utilities.

Theorem 11. *Every perfect strategy is a Nash equilibrium.*

Before presenting the proof of the theorem we start with some definitions and preliminary results.

Definition 19 (Desirability of a player)**.** *Desirability of a player will be the desirability of their pool. If they do not have one, their desirability will be the desirability of a hypothetical pool with their cost, the margin they have chosen and their personal stake.*

Note that for uniformity in this subsection we assume that all the players decide a margin even if they do not create a pool. In addition, when we rank the pools we take into account also the hypothetical pools described above. Ties break in favor of potential profit.

In Subsection 5.12.3 we define a more refined version of our game that consists of two stages and we remove these assumptions; we do not take into account non active pools in the ranking because we consider their desirability as zero and ties in ranking can break arbitrarily.

The following lemma is very useful and its proof follows directly from the definition of the reward function.

Lemma 8. *The quantity* $(r(x,s_j)-c_j)/x$ *as a function of* x *is increasing in* $(0,\beta)$ *and, if it is positive, decreasing in* (β,∞)*. Its maximum is achieved at* $x=\beta$*.*

The following lemma gives an upper bound on the utility of pool members. We will give an equilibrium that matches this upper bound.

Lemma 9. *In every joint strategy in which some player* j *is not a pool leader, their utility is at most* $\max_l D_l \cdot (s_j/\beta)$*, where* $\max_l D_l$ *is the maximum desirability among all players.*

Proof. It suffices to show that player j gets at most $D_l \frac{a_{j,l}}{\beta}$ from every pool l. The lemma follows directly from this by summing for all l: $\sum_l D_l \frac{a_{j,l}}{\beta} \leq \max_l D_l \sum_l \frac{a_{j,l}}{\beta} = \max_l D_l \frac{s_j}{\beta}$. The argument that for every pool l, player j gets at most $D_l \frac{a_{j,l}}{\beta}$ follows directly from the definition of the utility of pool members when we consider the two cases depending on whether rank_l is at most k and more than k.

Specifically, when $\text{rank}_l \leq k$, by the definition of the utility of pool members, the utility to player j from pool l is $D_l a_{j,l}/\sigma_l^{NM} \leq D_l a_{j,l}/\beta$.

When $\text{rank}_l > k$, its utility is given by

$$\begin{aligned}
&(1-m_l)\left(r(\lambda_l+a_{j,l},\lambda_l)-c_l\right)^+ \frac{a_{j,l}}{\lambda_l+a_{j,l}} \\
&\leq (1-m_l)\left(r(\beta,\lambda_l)-c_l\right)^+ \frac{a_{j,l}}{\beta} \\
&= D_l \frac{a_{j,l}}{\beta},
\end{aligned}$$

where the inequality comes from Lemma 8. □

We are now ready to present the proof of the Theorem.

Proof. (of Theorem 11) We first consider the simplified setting where players are mutually exclusively pool leaders or pool members.

Consider first a player j with rank at most k. This player is a pool leader of a pool of size β. We show that none of the possible responses improves their utility:

- Suppose that the player decreases their margin. This increases their desirability so that the new rank is still one of the first k ranks. Since the non-myopic stake remains the same, this move will decrease the utility of the player.

- Suppose that the player increases their margin. Since before the change the first $k+1$ players have the same desirability, the player's desirability drops and the rank becomes larger than k. As a result the player will be alone in a pool and their utility can only decrease (Lemma 8).

- Suppose that the player becomes a pool member of other pools. By Lemma 9, their utility can be $P(s_{k+1}, c_{k+1})s_j/\beta$ at most, which is lower that their current utility by $P(s_j, c_j) - P(s_{k+1}, c_{k+1})$ (by Equation 5.3).

We now consider a player j with rank higher than k. Again we show that none of the possible responses improves their utility. Notice first that by changing their allocation of stake, it can only hurt their utility since some of their stake ends up in pools with stake different than β, which can only lower their utility by Lemma 8. The other alternative is that the player becomes a pool leader. Since their rank is higher than k, the (non-myopic) stake of the pool contains only their own stake, which by Lemma 8 is again no better than the current utility.

We now sketch the full argument that considers the more complex strategies of possibly simultaneously delegating and creating a pool for each player (we remark that this case is also subsumed in the two-stage game in Subsection 5.12.3). Note that the desirability and thus the rank of the pools does not depend on the size of the pools. So if we allow strategies where a player is pool leader and simultaneously delegates some stake to other pools, then the perfect strategies remain Nash equilibria. In addition, it is easily verified that Lemmas 8,9 hold also in this case.

- If a player $\in \{1, ..., k\}$ with stake s and cost c increases their margin from m^* to m' and delegates stake $s - \lambda$ to other pools then their pool will have

rank higher than k and their utility will become at most $\frac{\lambda}{\lambda} \cdot (r(\lambda,\lambda) - c) + P(s_{k+1}, c_{k+1}) \cdot \frac{s-\lambda}{\beta}$ which is no higher than $\frac{\lambda}{\beta} \cdot P(\lambda, c) + P(s_{k+1}, c_{k+1}) \cdot \frac{s-\lambda}{\beta}$ because $\frac{r(\sigma,\lambda)-c}{\sigma}$ increasing for $\sigma \leq 1/k$ (Lemmas 8). This is at most $(m^* + (1-m^*) \cdot \frac{\lambda}{\beta}) \cdot P(s,c) + P(s_{k+1}, c_{k+1}) \cdot \frac{s-\lambda}{\beta}$ that is equal to their current utility.

- If a player $\in \{1, ..., k\}$ with stake s and cost c decreases their margin from m^* to m and simultaneously transfers stake $s - \lambda$ to other pools, then the desirability of their pool remains the same, increases or decreases. We will prove that in all cases their utility will be at most their current utility $(m^* + (1-m^*) \cdot \frac{s}{\beta}) \cdot P(s,c)$.

    1. If the desirability of their pool remains the same, then (i) the utility for the part of their stake that remains in their pool denoted by λ will decrease because of the lower margin or will remain the same and (ii) the utility for the stake that has been transferred to other pools denoted by $s - \lambda$ will also decrease because these pools have the same desirability and their non-myopic stake will become higher than $1/k$.

    2. If the desirability of their pool decreases, then the rank of their pool will become higher than k regardless the stake this player delegated to other pools. So again the utility for both parts of stake will decrease.

    3. If the desirability of their pool increases then their utility will become at most $(m + (1-m) \cdot \frac{\lambda}{\beta}) \cdot P(\lambda, c) + \frac{s-\lambda}{\beta} \cdot P(s_{k+1}, c_{k+1}) \leq (m^* + (1-m^*) \cdot \frac{\lambda}{\beta}) \cdot P(s,c) + \frac{s-\lambda}{\beta} \cdot P(s_{k+1}, c_{k+1}) = (m^* + (1-m^*) \cdot \frac{s}{\beta}) \cdot P(s,c)$.

- If a player $\in \{1, ..., k\}$ with stake s and cost c does not change margin and transfers stake $s - \lambda$ to other pools then again their utility will become at most $\frac{\lambda}{\lambda} \cdot (r(\lambda,\lambda) - c) + P(s_{k+1}, c_{k+1}) \cdot \frac{s-\lambda}{\beta}$ because their pool will have rank higher than k.

- If a player $\in \{k+1, ..., n\}$ with stake s and cost c creates a pool with stake λ and delegates the remaining stake to other pools then their pool will have rank lower than k so their utility will be at most $(r(\lambda,\lambda) - c) + \frac{s-\lambda}{\beta} \cdot P(s_{k+1}, c_{k+1}) \leq P(\lambda, c) \cdot \frac{\lambda}{\beta} + \frac{s-\lambda}{\beta} \cdot P(s_{k+1}, c_{k+1})$ which is not higher than their current utility $\frac{s}{\beta} \cdot P(s_{k+1}, c_{k+1})$.

□

5.12.2 Sybil Resilience

In this subsection we prove that our RSS is Sybil resilient in the sense that we provide a lower bound of the stake needed for Sybil behavior to be effective that is linear in the number of the identities. In addition, our analysis in this subsection captures also the scenario of a "whale" stakeholder with stake more than $1/k$ (which enables us to examine what happens when we remove the assumption that each stakeholder has stake at most $1/k$).

We consider an extended setting that involves a set of $\tilde{n} \leq n$ agents, each one with (private) stake $\tilde{s}_1, \ldots, \tilde{s}_{\tilde{n}}$ and associated (private) cost $\tilde{c}_1, \ldots, \tilde{c}_{\tilde{n}}$. Each agent i can declare themselves as a single player in the stake-pool game as long as $\tilde{s}_i \leq \beta$, or alternatively declare more than one players (called *Sybils*)[43] splitting their stake in some way between the declared players. This "pre-game" stage defines a specific instance of the stake-pool game. The utility of each agent is the sum of the utility of all the players that the agent controls.

We analyze two scenarios in this setting. In the first one, there is a utility non-maximizer agent with total stake less than $1/2$, who creates $k/2$ players, potentially lying about their costs, with the objective of dominating the system by creating $k/2$ saturated pools at the Nash equilibrium. In the second scenario, a utility maximizer agent creates $t > 1$ players that share their costs by using the same server. In both cases, to simplify the analysis, we will assume that the stake-pool game proceeds with players acting rationally and independently.

For a given agent, denote by $A \subseteq \{1, \ldots, n\}$ the set of players the agent introduces in the stakepool game. For each A, we denote by (s_i^A, c_i^A) the stake and cost of the i-th player in the game, ordering them in decreasing order of potential profit, excluding A. Moreover, the maximum cost and the minimum stake, excluding players in A, will be denoted $c_{\max}^A$ and $s_{\min}^A$ respectively. We prove the following.

Theorem 12. *Consider an agent controlling a set of players A. First, if the agent has stake less than $\frac{k}{2} \cdot \left(s_{k/2+1}^A - \frac{c_{\max}^A}{R} \cdot (1 + \frac{1}{\alpha}) \right)$ then it will control fewer than $k/2$ saturated pools at the Nash equilibrium, even if the agent is a utility non-maximizer. Second, if the agent is a utility maximizer with cost $\tilde{c}$ and stake less than $t \cdot \left(s_{k-t+1}^A - \frac{(c_{\max}^A - \tilde{c}/t)}{R} \cdot (1 + \frac{1}{\alpha}) \right)$, it will control fewer than t saturated pools at the Nash equilibrium for any $k \geq t > 1$.*

Proof. Consider the the equilibrium of Theorem 11, where the k players with the

highest potential profit $P(s_i, c_i)$ have a saturated pool, cf. Proposition 2.

Let n be the number of players in the stake-pool game, some of which in a subset A are Sybil players controlled by an agent with stake $\tilde{s}$ and cost $\tilde{c}$. For simplicity we drop the superscript A from s_i^A, c_i^A.

Let us suppose first that the agent creates $t > 1$ Sybil players with stake $s = \tilde{s}/t$ each, and claims cost c for each one equal to (i) $\frac{c_{\text{fake}}}{k/2}$, in the first case (where $t = k/2$) where c_{fake} is some arbitrary cost potentially below $\tilde{c}$, and (ii) $\tilde{c}/t$, in the second case. The objective of the agent in the first case is to create $k/2$ saturated pools so that it musters the highest possible influence in the system, and in the second case to maximize its utility by sharing the same server for all its Sybil players. We provide a lower bound in both cases for the stake the agent needs in order to create t saturated pools. After that, we generalize the above result by allowing the attacker to split its stake and cost arbitrarily among the Sybil players. Whenever needed, without loss of generality, we will break any ties in favor of the adversary. Firstly we will prove a lemma regarding when the attacker succeeds in creating t pools when the Sybil players are identical and play rationally without colluding.

Lemma 10. *An attacker with t identical Sybil players, with stake s and cost c each one, controls t saturated pools at the Nash equilibrium with rank at most k if and only if $P(s,c) \geq P(s_{k-t+1}, c_{k-t+1})$*

Proof.

- ($\Leftarrow$) If $P(s,c) \geq P(s_{k-t+1}, c_{k-t+1})$ then at most $k-t$ players may have higher $P(s_i, c_i)$ value than the Sybil players and thus all the Sybil players will have a saturated pool.

- ($\Rightarrow$) if $P(s,c) < P(s_{k-t+1}, c_{k-t+1})$ then there would exist at least $k-t+1$ players with $P(s_i, c_i)$ higher than the agent's players hence superseding all the players in the pool rankings. Thus at most $t-1$ Sybil players can have a saturated pool.

This completes the proof of the lemma. □

Now we observe that

$$P(s,c) \geq P(s_{k-t+1}, c_{k-t+1}) \iff s \geq s_{k-t+1} - \frac{1}{R} \cdot (c_{k-t+1} - c) \cdot (1 + \frac{1}{\alpha}) \tag{5.4}$$

Finally, with respect to the two scenarios, we have the following. In the first scenario, the attacker does not care about cost and hence can set $c = 0$.

$\tilde{s} < \frac{k}{2} \cdot (s_{k/2+1} - \frac{c_{\max}}{R} \cdot (1 + \frac{1}{\alpha}))$, we obtain

$$s = \tilde{s}/t < s_{k/2+1} - \frac{c_{\max}}{R} \cdot (1 + \frac{1}{\alpha}) \leq s_{k-t+1} - \frac{1}{R} \cdot (c_{k-t+1} - c) \cdot (1 + \frac{1}{\alpha})$$

by setting $t = k/2$ and observing that $c_{k-t+1} - c \leq c_{\max}$, from which we obtain that $P(s,c) < P(s_{k/2+1}, c_{k/2+1})$ and hence the adversary fails to control $k/2$ pools in the equilibrium.

In the second scenario where it holds that $c = \tilde{c}/t$, and

$$\tilde{s} < t \cdot \left(s_{k-t+1} - \frac{(c_{\max} - \tilde{c}/t)}{R} \cdot (1 + \frac{1}{\alpha}) \right)$$

we obtain

$$s = \tilde{s}/t < s_{k-t+1} - \frac{(c_{\max} - \tilde{c}/t)}{R} \cdot (1 + \frac{1}{\alpha}) \leq s_{k-t+1} - \frac{1}{R} \cdot (c_{k-t+1} - c) \cdot (1 + \frac{1}{\alpha})$$

recalling that $c = \tilde{c}/t$ and $c_{\max} \geq c_{k-t+1}$. It follows that $P(s,c) < P(s_{k-t+1}, c_{k-t+1})$ and hence the adversary fails to control t pools in the equilibrium.

In order to generalize the above results so that they hold even if the stake and the cost of the attacker is split arbitrarily among the Sybil players we will prove by contradiction the following lemma.

Lemma 11. *If an agent splits its stake and cost among the Sybil players so that it succeeds in creating t saturated pools at the Nash equilibrium then it would succeed even if the split of stake and cost is done equally among the Sybil players.*

Proof. We will prove the theorem by contradiction. We will assume that the attacker splits its stake and cost among the Sybil players so that it has t saturated pools at the Nash equilibrium but if it splits them equally then it won't. Let $\tilde{s}_1, ..., \tilde{s}_t$ and $\tilde{c}_1, ..., \tilde{c}_t$ the stake and the cost of the Sybil players. It holds $\tilde{s}_1 + ... + \tilde{s}_t = \tilde{s}$ and $\tilde{c}_1 + ... + \tilde{c}_t \leq \tilde{c}$ ("=" in the case of a utility maximizing agent).

If the attacker has t saturated pools at the Nash equilibrium, then player $k-t+1$ will not have a pool. If the $(k-t+1)$-th player had a pool, then also players $1, 2..., k-t$ would have pools and the attacker would have at most $t-1$ pools. Recall that we break ties in favor of the attacking agent. As a result $P(s_{k-t+1}, c_{k-t+1})$ is smaller or equal than the potential profit of all the Sybil players (recall that all the Sybil players have a pool and that at the Nash equilibrium the k players with the highest potential profit have a pool), otherwise $k-t+1$ would have a pool. Thus it holds by equation 5.4 that

$\forall i : \tilde{s}_i \geq s_{k-t+1} - \frac{1}{R} \cdot (c_{k-t+1} - \tilde{c}_i) \cdot (1 + \frac{1}{\alpha})$. Summing for all i, we obtain that $\tilde{s} \geq tX + \frac{1}{R}(1 + \frac{1}{\alpha}) \sum_{i=1}^{t} \tilde{c}_i$, where $X = s_{k-t+1} - \frac{1}{R} \cdot c_{k-t+1} \cdot (1 + \frac{1}{\alpha})$.

If, on the other hand, the agent splits its stake equally and declares the same cost c', for some $c' \geq 0$, for all Sybil players but it does not succeed in having t saturated pools then by Lemma 10 and equation 5.4 we have $\tilde{s}/t < s_{k-t+1} - \frac{1}{R} \cdot (c_{k-t+1} - c') \cdot (1 + \frac{1}{\alpha})$, which implies $\tilde{s} < t \cdot X + t \cdot \frac{1}{R}(1 + \frac{1}{\alpha})c'$. Combining the above constraints on $\tilde{s}$, we obtain that $\sum_{i=1}^{t} \tilde{c}_i < t \cdot c'$.

In the case of a utility maximizer, we have that $\sum_{i=1}^{t} \tilde{c}_i = \tilde{c}$ and since we can set $c' = \tilde{c}/t$, we obtain a contradiction. In the case of a utility non-maximizer, on the other hand, we can set $c' = \sum_{i=1}^{t} \tilde{c}_i / t$, hence also obtaining a contradiction. □

Combining the above results, we arrive at the proof of the theorem.

□

Analysing the bound: we observe that in both cases, the minimum stake needed by the Sybil attacker agent is asymptotically linear in the number of stake pools ($k/2$ in the first case and t in the second).

Moreover, the coefficient, in both cases, can be adjusted by varying the Sybil resilience parameter α. Specifically, when $\frac{c^A_{\max}}{R} < s^A_{\min}$, these bounds are positive for suitable value of α; in particular, the higher α is, the higher these bounds become. Note that $s^A_{k/2+1}$ and s^A_{k-t+1} are nondecreasing in α, because the ordering of the remaining agents depends on $P(s_i, c_i)$ and thus also in α (the higher α is the higher impact agents' stake has on the ordering). For example, in the first case when $R = 1$ and $k = 10$, and the stake and cost are sampled from a Pareto distribution with parameter $\alpha = 2$ and the uniform distribution from $[0.0005, 0.0010]$ respectively, if we choose $\alpha = 0.5$ then $c^A_{k/2+1} = 0.00076024, s^A_{k/2+1} = 0.02002176$. Then if a utility non-maximizer declares cost $c = 0.9 \cdot c^A_{k/2+1}$, the stake required for the attack is at least 0.0989. This is not far from optimal, since the largest possible lower bound is $5 \cdot 0.02002176 = 0.1001088$, which would apply to the setting of negligible costs and a choice of α that goes to $+\infty$.

How possible is that there is a stakeholder with stake as much as this lower bound? We examine the probability under reasonable probability distributions that there exists an agent who has stake more than $\frac{k}{2} \cdot s^A_{k/2+1}$, which allows them to engage in Sybil behavior in the above settings (i.e., with negligible costs and a choice of α that goes to $+\infty$).

Let S_i and $s_i = \frac{S_i}{\sum_{i=1}^{\tilde{n}} S_i}$ be the absolute and the relative stake respectively of agent i. Let $S_1, ..., S_{\tilde{n}}$ be independent samples from random variable X that follows the upper truncated Pareto distribution [37] with parameter $a \neq 0$. Let θ and T be the minimum and maximum value of the distribution, respectively. Then the cumulative function of X is $F_X(x) = \frac{1-(\frac{\theta}{x})^a}{1-(\frac{\theta}{T})^a}$ when $\theta \leq x \leq T$. Also if X_r is the stake of the agent with the r-th smallest stake, then the cumulative function of X_r is $F_{X_r}(x) = \sum_{j=r}^{\tilde{n}} [\binom{\tilde{n}}{j} \cdot F_X^j(x) \cdot (1 - F_X(x))^{\tilde{n}-j}]$, see [35]. We also denote by $S_i = X_{\tilde{n}-i+1}$ the stake of the agent with the i-th highest stake. Let $f_{S_{\frac{k}{2}+1}}(t)$ be the density function of $S_{\frac{k}{2}+1}$ and $F_B(k; \tilde{n}, p) = \sum_{i=0}^{k} \binom{\tilde{n}}{i} \cdot p^i \cdot (1-p)^{\tilde{n}-i}$ be the cumulative function of Binomial distribution. The following theorem quantifies the probability that a Sybil attack is possible.

Theorem 13. *Assume that $S_1,\ldots,S_{\tilde{n}}$, where S_i is the absolute stake of agent i, are drawn from an upper truncated Pareto distribution with parameters a,θ,T. Then when $\delta = \left(\frac{1-(\frac{\theta}{T})^a}{1-(\frac{\theta\cdot k}{2\cdot T})^a}\right)\cdot(1-\frac{k}{2\tilde{n}})-1>0$ we have $Pr(s_1 > \frac{k}{2}\cdot s_{\frac{k}{2}+1}) \leq e^{-\delta^2\mu/3}$, where $\mu = \tilde{n}\cdot F_X(\frac{2T}{k})$.*

Proof. Let A be the event $S_1 > \frac{k}{2}\cdot S_{\frac{k}{2}+1}$ and X_B a random variable that follows Binomial distribution with parameters n and $F_X(\frac{2T}{k})$. By the Continuous Law of Alternatives we have [2]:

$$\begin{aligned}
Pr(S_1 > \frac{k}{2}\cdot S_{\frac{k}{2}+1}) &\approx \int_{-\infty}^{+\infty} Pr(A/S_{\frac{k}{2}+1}=t)\cdot f_{S_{\frac{k}{2}+1}}(t)dt \\
&= \int_{\theta}^{\frac{2T}{k}} Pr(A/S_{\frac{k}{2}+1}=t)\cdot F'_{S_{\frac{k}{2}+1}}(t)dt \\
&= \int_{\theta}^{\frac{2T}{k}} Pr(S_1 > \frac{k}{2}\cdot t)\cdot F'_{S_{\frac{k}{2}+1}}(t)dt \\
&= \int_{\theta}^{\frac{2T}{k}} (1-F_{S_1}(\frac{k}{2}\cdot t))\cdot F'_{S_{\frac{k}{2}+1}}(t)dt \\
&= \int_{\theta}^{\frac{2T}{k}} (1-F_X^n(\frac{k}{2}\cdot t))\cdot F'_{S_{\frac{k}{2}+1}}(t)dt \\
&\leq \int_{\theta}^{\frac{2T}{k}} F'_{S_{\frac{k}{2}+1}}(t)dt \\
&= F_{S_{\frac{k}{2}+1}}(\frac{2T}{k}) - F_{S_{\frac{k}{2}+1}}(\theta) \\
&= F_{S_{\frac{k}{2}+1}}(\frac{2T}{k}) \\
&= F_{X_{\tilde{n}-\frac{k}{2}}}(\frac{2T}{k}) \\
&= \sum_{j=\tilde{n}-\frac{k}{2}}^{\tilde{n}} [\binom{\tilde{n}}{j}\cdot F_X^j(\frac{2T}{k})\cdot\left(1-F_X(\frac{2T}{k})\right)^{\tilde{n}-j}] \\
&= Pr(X_B \geq \tilde{n}-\frac{k}{2}) \\
&= Pr(X_B \geq (1+\delta)\cdot\mu) \\
&\leq e^{-\delta^2\mu/3}
\end{aligned}$$

where $\delta = \left(\frac{1-(\frac{\theta}{T})^\alpha}{1-(\frac{\theta\cdot k}{2\cdot T})^\alpha}\right)\cdot(1-\frac{k}{2\tilde{n}})-1>0$ and $\mu = \tilde{n}\cdot F_X(\frac{2T}{k})$. □

Note that if we take $a=1$, $\frac{\theta}{T}=1/100,000$ and $k=100$, then in order for δ to be positive, it suffices $\tilde{n}>150,000$, a reasonable number of users of a general

[2] Let A be an event, X a continuous random variable and f_X the density function then it holds $Pr(A)\approx\int_{-\infty}^{+\infty} Pr(A/X=t)\cdot f_X(t)dt$ (see Section 6.4 in [15]).

cryptocurrency. Also if we choose higher $\tilde{n}$ or θ and lower T, then δ will increase. It holds that δ is

- increasing as a function of $\tilde{n}$ and decreasing as a function of T and a
- increasing as a function of k if and only if $\frac{\theta^a \cdot k^{a-1}}{2^a \cdot T^a} \cdot (k + 2 \cdot a \cdot \tilde{n} - a \cdot k) > 1$. In particular when $a = 1$, δ is increasing as a function of k if and only if $T < \theta \cdot \tilde{n}$.

5.12.3 The Two-Stage Stake Pool Game and its Equilibria

We next prove that our reward sharing scheme effectively retains the same equilibria outcome of Theorem 11 also in a more realistic two-stage or "inner-outer game." The advantages of this approach are as follows: (i) it allows us to analyze non-myopic moves in response to pool leaders changing margin or allocation, (ii) in this setting when a pool has not been activated, we define its desirability to be zero, something that gives us a more realistic result, because in practice only pools that have already been created will be ranked, (iii) in this game we break ties in ranking in arbitrary ways, not only according to potential profit. We note that similar non-myopic type of play has already been considered in other settings, notably in *Cournot Equilibria* [55], as we will discuss in the related work.

5.12.3.1 The Inner-outer Game and the Strategies

Our "inner-outer game" consists of two games. In the *outer* game, player i decides on the margin m_i and on the stake λ_i to be allocated to its own pool, in case the player will decide to activate it in the inner game. So a strategy of a player i in the outer game is a tuple (m_i, λ_i) of margin and allocated stake, and let $(\vec{m}, \vec{\lambda})$ be the joint strategy of the outer game. Each joint strategy of the outer game determines one inner game.

In the *inner* game, the margins $\vec{m}$ and the stakes $\vec{\lambda}$, which potential pool leaders would allocate to their pools, are given, and the strategies of the players are their allocations. So in the inner game determined by $(\vec{m}, \vec{\lambda})$, a strategy of player i is $S_i^{(\vec{m},\vec{\lambda})} = \vec{a}_i$, and a joint strategy is $\vec{S}^{(\vec{m},\vec{\lambda})}$. Note that if a player i decides to activate its own pool, which means $a_{i,i} > 0$, then the player is committed to allocate stake λ_i to its own pool, where λ_i is part of the strategy of the outer game. So $a_{i,i} \in \{0, \lambda_i\}$.

5.12.3.2 Definition of Equilibria for Inner and Outer Game and Utilities

Definition 20. *A joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$ is a Nash equilibrium of the inner game defined by $(\vec{m},\vec{\lambda})$ when for every player j*

$$u_j(S_j'^{(\vec{m},\vec{\lambda})},\vec{S}_{-j}^{(\vec{m},\vec{\lambda})}) \leq u_j(\vec{S}^{(\vec{m},\vec{\lambda})}) \tag{5.5}$$

for every $S_j'^{(\vec{m},\vec{\lambda})} \neq S_j^{(\vec{m},\vec{\lambda})}$. This is the standard Nash equilibrium notion when the players try to maximize their non-myopic utility cf. Definition 17.

To define the non-myopic equilibrium of the outer game, let us temporarily assume that there is a *unique Nash equilibrium in every inner game*. Then we define the utility of player j in the outer game, where players have selected joint strategy $(\vec{m},\vec{\lambda})$, as: $u_j^{\text{outer}}(\vec{m},\vec{\lambda}) = u_j(\vec{S}^{(\vec{m},\vec{\lambda})})$, where $\vec{S}^{(\vec{m},\vec{\lambda})}$ is the unique equilibrium of the inner game determined by $(\vec{m},\vec{\lambda})$. So a joint strategy $(\vec{m},\vec{\lambda})$ is an approximate ε-non-myopic Nash equilibrium of the outer game when for every player j

$$u_j^{\text{outer}}(m_j',\vec{m}_{-j},\lambda_j',\vec{\lambda}_{-j}) \leq u_j^{\text{outer}}(\vec{m},\vec{\lambda}) + \varepsilon \tag{5.6}$$

for every $(m_j',\lambda_j') \neq (m_j,\lambda_j)$.

When there are multiple equilibria in the inner game, we define $u_j^{\text{outer}}(\vec{m},\vec{\lambda})$ as *the set* of values $u_j(\vec{S}^{(\vec{m},\vec{\lambda})})$, where $\vec{S}^{(\vec{m},\vec{\lambda})}$ is a Nash equilibrium of the inner game determined by $(\vec{m},\vec{\lambda})$.

Let

$$u_j^{\text{outer,up}}(\vec{m},\vec{\lambda}) = \begin{cases} \sup u_j^{\text{outer}}(\vec{m},\vec{\lambda}) & \text{if } u_j^{\text{outer}}(\vec{m},\vec{\lambda}) \neq \emptyset, \\ -\infty & \text{elsewhere.} \end{cases} \tag{5.7}$$

In the same way we define:

$$u_j^{\text{outer,low}}(\vec{m},\vec{\lambda}) = \begin{cases} \inf u_j^{\text{outer}}(\vec{m},\vec{\lambda}) & \text{if } u_j^{\text{outer}}(\vec{m},\vec{\lambda}) \neq \emptyset, \\ -\infty & \text{elsewhere.} \end{cases} \tag{5.8}$$

Note that when $u_j^{\text{outer}}(\vec{m},\vec{\lambda})$ is not empty, it is a non-empty bounded subset of the reals and therefore always has both supremum and infimum: upper and lower bounds are given by R and $(-\max\{c_1,\ldots,c_n\})$ respectively.

Definition 21. *A joint strategy $(\vec{m},\vec{\lambda})$ is an ε-non-myopic Nash equilibrium when for every player j*

$$u_j^{\text{outer},up}(m_j',\vec{m}_{-j},\lambda_j',\vec{\lambda}_{-j}) \leq u_j^{\text{outer},low}(\vec{m},\vec{\lambda}) + \varepsilon \tag{5.9}$$

for every $(m_j',\lambda_j') \neq (m_j,\lambda_j)$.

5.12.3.3 Summary of our Results and a Basic Lemma

Now we describe a set of joint strategies that (i) are approximate non-myopic Nash equilibria of the outer game and (ii) have the characteristic that in the inner games defined by these joint strategies, all the equilibria form k saturated pools. Recall that a pool is *saturated* when its stake is at least β. The pool leaders of these pools in these equilibria of the inner game are again the players with the highest values $P(s_i, c_i)$. Note that if all players activated a pool of size $1/k$ with the same margin and their whole stake, then the k pools with the highest potential profit ($P(s_i, c_i)$) would give the highest rewards to their members.

The intuition for how the set of margins of these joint strategies is determined is the following: the k players with the highest values $P(s_i, c_i)$ set the maximum margin they can, so that their pools belong to the k most desirable pools (the pools with the highest desirability), no matter which margins the other players set. Note that in this model, these players want their pools to have strictly higher desirability than the potential pool of player $k+1$, because in a tie in ranking, they might otherwise lose.

In more detail: let $G = \{1, \ldots, k\}$ be the set of those k players with the highest $P(s_i, c_i)$, $\varepsilon = P(s_k, c_k) - P(s_{k+1}, c_{k+1})$ and $\varepsilon_1 = P(s_{k+1}, c_{k+1}) - P(s_{k+2}, c_{k+2})$. By assumption $\varepsilon, \varepsilon_1 > 0$.

For ε' such that $0 < \varepsilon' < \min\{\varepsilon, P(s_{k+1}, c_{k+1})\}$ and α such that $\frac{s_{k+1}}{\beta} < \alpha < 1$, we define $\vec{m}^*(\varepsilon', \alpha)$ as follows:

$$m_i^*(\varepsilon', \alpha) = \begin{cases} \frac{P(s_i,c_i) - P(s_{k+1},c_{k+1}) - \varepsilon' \cdot (1-\alpha)}{P(s_i,c_i)} & \text{when } i \in G, \\ \frac{\varepsilon' \cdot \alpha}{P(s_{k+1},c_{k+1})} & \text{when } i = k+1, \\ 0 & \text{elsewhere,} \end{cases}$$

and let $\vec{\lambda}^*$ be the vectors with $\lambda_i^* = s_i$ for $i \in [n]$.

Note that margins are well defined because $0 \leq m_i^* < 1$. We prove that:

- for each ε' such that $0 < \varepsilon' < \min\{\varepsilon, P(s_{k+1}, c_{k+1}), \varepsilon_1\}$, the joint strategies

$$(\vec{m}^*(\varepsilon', \alpha), \vec{\lambda}^*)_{\frac{s_{k+1}}{\beta} < \alpha < 1}$$

  are ε'-non-myopic Nash equilibria of the outer game. So we prove that for every ε' as defined above, there is a class of joint strategies that are ε'-non-myopic Nash equilibria of the outer game (Theorem 15).

- for each ε' such that $0 < \varepsilon' < \min\{\varepsilon, P(s_{k+1}, c_{k+1}), \varepsilon_1\}$ and α such that $\frac{s_{k+1}}{\beta} < \alpha < 1$, all the equilibria of the inner game determined by joint strategy $(\vec{m}^*(\varepsilon', \alpha), \vec{\lambda}^*)$ form k saturated pools (Theorem 14).

- the pool leaders of the k saturated pools described above are the players of G, which are the players with the highest $P(s,c)$ (Theorem 14).

First, we will state a basic lemma that we use in the proofs of the following lemmas and theorems. This lemma is a generalization of Lemma 9 and intuitively says that according to any joint strategy of any inner game, the utility that a player takes from allocating stake s to other active pools is upper bounded by $D_{\max} \cdot s/\beta$, where $D_{\max}$ is the maximum desirability of all the other active pools. Formally:

Lemma 12. *For every joint strategy of the outer game $(\vec{m}, \vec{\lambda})$ and for every joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$ of the inner game determined by $(\vec{m}, \vec{\lambda})$, it holds: For every player j that has allocated stake s to other active pools $\sum_{i \in [n] \setminus \{j\} : i \text{ active}} u_{j,i}(\vec{S}^{(\vec{m},\vec{\lambda})}) \leq D_{\max} \cdot \frac{s}{\beta}$, where $D_{\max}$ is the maximum desirability according to $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$ of all the other active pools.*

Its proof is similar to the proof of Lemma 9. Recall that inactive pools in this model have desirability zero and their members (if they exist) take utility zero from these pools.

Proof. Note that according to any joint strategy of any inner game, the utility of a player from a pool to which it has allocated stake s is upper bounded by $(1-m) \cdot \frac{s}{\beta} \cdot (r(\beta, \lambda) - c)^+$, where m is the margin, c is the cost and λ is the stake that pool leader has allocated to this pool. This holds because of the following two properties of the rewards scheme.

1. when $\sigma < \beta$ it holds: $\frac{d[r(\sigma,\lambda)-c)\cdot\frac{1}{\sigma}]}{d\sigma} > 0$.

2. $\forall \lambda \; r(\sigma, \lambda) = r(\beta, \lambda)$, when $\sigma > \beta$, which means that $\frac{d[r(\sigma,\lambda)-c)\cdot\frac{1}{\sigma}]}{d\sigma} < 0$ for $\sigma > \beta$ when $P(\lambda, c) > 0$.

So we take an arbitrary strategy of the outer game $(\vec{m}, \vec{\lambda})$, an arbitrary joint strategy $\vec{S}^{(\vec{m},\vec{\lambda})}$ of the inner game determined by $(\vec{m}, \vec{\lambda})$ and an arbitrary player j

that has allocated stake s to other active pools and we have:

$$\begin{aligned}\sum_{i\in[n]\setminus j:i\text{active}} u_{j,i}(\vec{S}^{(\vec{m},\vec{\lambda})}) &\leq \sum_{i\in[n]\setminus j:i\text{active}} [(1-m_i)\cdot\frac{a_{j,i}(\vec{S}^{(\vec{m},\vec{\lambda})})}{\beta}\cdot(r(\beta,\lambda_i)-c_i)^+] \\ &= \sum_{i\in[n]\setminus j:i\text{active}} [D_i(\vec{S}^{(\vec{m},\vec{\lambda})})\cdot\frac{a_{j,i}(\vec{S}^{(\vec{m},\vec{\lambda})})}{\beta}] \\ &\overset{D_i(\vec{S}^{(\vec{m},\vec{\lambda})})\leq D_{\max},\sum_{i\in[n]\setminus j} a_{j,i}=s}{\leq} D_{\max}\cdot\frac{s}{\beta}.\end{aligned}$$

□

5.12.3.4 Equilibria of the Inner Game

Theorem 14. *For every ε': $0<\varepsilon'<\min\{\varepsilon,P(s_{k+1},c_{k+1}),\varepsilon_1\}$ and for every α: $\frac{s_{k+1}}{\beta}<\alpha<1$, it holds: A joint strategy $\vec{S}^{(\vec{m}^*(\varepsilon',\alpha),\vec{\lambda}^*)}$ of the inner game determined by $(\vec{m}^*(\varepsilon',\alpha),\vec{\lambda}^*)$ is a Nash equilibrium if and only if it forms k active, saturated pools, whose pool leaders belong to G.*

Proof. This can be proved by the following two Lemmas 13, 14. □

Lemma 13. *For every ε': $0<\varepsilon'<\min\{\varepsilon,P(s_{k+1},c_{k+1},\varepsilon_1\}$ and for every α: $\frac{s_{k+1}}{\beta}<\alpha<1$, it holds: In an inner game determined by $(\vec{m}^*(\varepsilon',\alpha),\vec{\lambda}^*)$, joint strategies $\vec{S}^{(\vec{m}^*(\varepsilon',\alpha),\vec{\lambda}^*)}$ that form k active saturated pools, whose pool leaders belong to G, are Nash equilibria.*

Proof. We take an arbitrary ε': $0<\varepsilon'<\min\{\varepsilon,P(s_{k+1},c_{k+1}),\varepsilon_1\}$ and an arbitrary α: $\frac{s_{k+1}}{\beta}<\alpha<1$ For simplicity we write $\vec{m}^*$ instead of $\vec{m}^*(\varepsilon',\alpha)$. Let us take an arbitrary joint strategy $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$ that forms k saturated active pools, whose pool leaders belong to G. This means that players in G have decided to activate their own pools with margins and the stakes determined by $(\vec{m}^*,\vec{\lambda}^*)$ in the outer game and that the other players have delegated all their stake to these pools, such that each pool has stake β. We will prove that this joint strategy is a Nash equilibrium of the inner game determined by $(\vec{m}^*,\vec{\lambda}^*)$, or in other words that no player can increase their non-myopic utility by changing their strategy, given that the other players do not change their strategies.

- Let us take a player j that belongs to G. Recall that $\lambda_j^*=s_j$. The way in which they can change their strategy is to dissolve their pool (make it inactive) and to allocate their stake to one or more other pools whose pool

leaders also belong to G or to leave it unallocated. Note that they can also allocate their stake to an inactive pool whose pool leader decided not to allocate the stake (determined in the outer game) to it, but in this case their utility will become zero. So we can assume that the player in this case leaves their stake unallocated.

1. Let us take a strategy $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$ of player j, according to which they dissolve their pool and allocates their stake to one or more other pools whose pool leaders also belong to G. Intuitively this strategy will decrease player j's utility, because they will lose the rewards from the margin of their own pool and additionally cannot increase their utility as a pool member of the other pools, because no pool has higher desirability than their own pool. Note that a player in G in the inner game determined by $(\vec{m}^*,\vec{\lambda}^*)$ cannot choose to activate their own pool and simultaneously be a member of another pool as their strategy, given that $\lambda^*_j = s_j$ and therefore $a_{j,j} \in \{0, s_j\}$. Formally: Regarding player j's current non-myopic utility, we have:

$$
\begin{aligned}
u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) &\overset{\lambda^*_i = s_i}{=} u_{j,j}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \\
&= (m^*_j + (1-m^*_j)\cdot\frac{s_j}{\beta})\cdot P(s_j,c_j) \\
&\overset{P(s_j,c_j)>0}{>} (1-m^*_j)\cdot\frac{s_j}{\beta}\cdot P(s_j,c_j) \\
&= \frac{s_j}{\beta}\cdot(P(s_{k+1},c_{k+1}) + \varepsilon'\cdot(1-\alpha)) > 0.
\end{aligned}
$$

Note that $\text{rank}_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \leq k$, because only k pools are active and π_j has positive desirability, given that $P(s_j,c_j) > P(s_{k+1},c_{k+1}) > 0$ by assumption. If they choose a different strategy $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$ as described above, where they allocate some part of their stake s_{j_1} to a pool π_l and the remaining part $s_{j_2} = s_j - s_{j_1}$ to a pool $\pi_{l'}$, then we have :

$$
\begin{aligned}
&u_{j,l}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \\
&= (1-m^*_l)\cdot P(s_l,c_l)\cdot\frac{s_{j_1}}{\beta+s_{j_1}} \\
&= (P(s_{k+1},c_{k+1}) + \varepsilon'\cdot(1-\alpha))\cdot\frac{s_{j_1}}{\beta+s_{j_1}}
\end{aligned}
$$

and

$$u_{j,l'}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)) \cdot \frac{s_{j_2}}{\beta + s_{j_2}}.$$

Note that by the description of $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$, both π_l and $\pi_{l'}$ have stake β, and their pool leaders belong to G. Furthermore, according to both joint strategies $(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})$ and $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$, at most k pools are active and have positive desirability (in $(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})$, player j has dissolved their pool), thus the ranking of all active pools is smaller or equal to k (they belong to the k most desirable pools). So

$$\begin{aligned} &u_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \\ &= (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)) \cdot \left(\frac{s_{j_2}}{\beta + s_{j_2}} + \frac{s_{j_1}}{\beta + s_{j_1}}\right) \\ &< u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}), \end{aligned}$$

which means that player j cannot increase their non-myopic utility by this change. Note that in the above proof, we did not assume anything specific about the stakes s_{j_1} and s_{j_2}. By induction on the number of the pools to which j allocates stake, we can prove in the same way that j's utility will not increase if they dissolve their pool. Note that the number of pools to which j allocates stake does not have an impact on desirability and ranking of the pools, because in all cases (i) there exist $k-1$ active pools with positive desirability (j has dissolved their own pool) and (ii) desirability, which determines ranking, does not depend on pool size.

2. Let us consider the case where j leaves their stake unallocated. Then their utility will drop to zero, which is smaller than their current utility $u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) > 0$.
3. Finally, let us look at the case where j leaves some of their stake unallocated and allocates the rest to the other existing pools. Then by using the inequalities of the two previous cases, we can prove that their utility will decrease.

- Consider a player $j \notin G$. According to $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$, this player has not activated their own pool. Then the ways in which they can change their strategy are

(i) to activate their own pool by allocating the stake and margin specified in the outer game, (ii) to remove some parts of their stake from some pools and allocate them to other already saturated pools, or (iii) to remove some parts of their stake from some pools and leave them unallocated or allocate them to inactive pools. Recall that again $\lambda_j^* = s_j$, so player j cannot activate their own pool and simultaneously allocate stake to other pools in the inner game determined by $(\vec{m}^*, \vec{\lambda}^*)$.

1. Consider the first case where j changes their strategy to $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$ by removing all their stake from the existing pools and activates their own pool as specified by the outer game. Intuitively, their utility will decrease, because if they activate their own pool, this pool will not belong to the k most desirable pools and thus will have non-myopic stake s_j. Additionally, the desirability of their own pool will be lower than the desirability of one of the active pools. Formally: their current utility is

$$\begin{aligned} u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) &= \sum_{i \in [n] \setminus j} u_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \\ &= \sum_{i \in [n] \setminus j} [(1 - m_i^*) \cdot \frac{a_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)})}{\beta} \cdot P(s_i, c_i)] \\ &= \sum_{i \in [n] \setminus j} [\frac{a_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)})}{\beta} (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1 - \alpha))] \\ &= \frac{s_j}{\beta} \cdot (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1 - \alpha)). \end{aligned}$$

   If they activate their own pool with their whole stake as specified by the outer game ($\lambda_i^* = s_i$), then the ranking $\text{rank}_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})$ of this pool π_j will be $k+1$. This holds because the desirability of π_j will be strictly smaller than the desirability of the k existing active pools, whose pool leaders belong to G.

   In more detail, if $j = k+1$, we have for all $i \in G$:

$$D_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})$$
$$= (1 - \frac{\varepsilon' \cdot \alpha}{P(s_{k+1}, c_{k+1})}) \cdot P(s_{k+1}, c_{k+1})$$
$$= P(s_{k+1}, c_{k+1}) - \varepsilon' \cdot \alpha < P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)$$
$$= D_i(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}).$$

If $j \neq k+1$, we have for all $i \in G$:

$$D_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \leq \big((1-0) \cdot P(s_j, c_j)\big)^+$$
$$< P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha) = D_i(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}).$$

As a result we have

$$u_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \overset{\lambda^*_j = s_j}{=} u_{j,j}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})$$
$$= \frac{R}{1+\alpha} \cdot [s_j + s_j \cdot \alpha \cdot \frac{s_j \cdot (1 - \frac{\beta - s_j}{\beta})}{\beta} - c_j \cdot \frac{1+\alpha}{R}]$$
$$\overset{\frac{d[(r(\sigma,\lambda)-c)\cdot\frac{1}{\sigma}]}{d\sigma} > 0, \text{when} \sigma < \beta \wedge j \notin G}{\leq} P(s_{k+1}, c_{k+1}) \cdot \frac{s_j}{\beta}$$
$$< (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)) \cdot \frac{s_j}{\beta}$$
$$= u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}),$$

which means that their utility will not increase by activating their own pool.

2. Now consider the case where j changes their strategy to $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$ by removing some parts of their stake from some pools and allocate them to two other, already saturated pools. We will prove that j's utility will not increase in this case. Then we can use induction on the number of pools to which they allocate stake according to their new strategy, so that we prove the following statement: their utility will not increase by removing parts of their stake from some pools and allocating them to other already saturated pools. Intuitively, their utility will decrease, because the reward function has the property $r(\sigma, \lambda) = r(\beta, \lambda)$ for $\sigma > \beta$.

Let s_{j_i} be the stake that j will remove from each pool π_i and $a'_{j,l}$, $a'_{j,l'}$ the stake that they will allocate to pools π_l and $\pi_{l'}$ respectively. Note that j's stake, that exists in a pool π_i after j removes s_{j_i}, is $a_{j,i}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = a_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) - s_{j_i}$ and that the stake that exists in the pools π_l and $\pi_{l'}$ after they allocate the stake removed from the other pools to them is $a_{j,l}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = a_{j,l}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) + a'_{j,l}$ and $a_{j,l'}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) = a_{j,l'}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) + a'_{j,l'}$ respectively. Their current utility is

$$\begin{aligned} u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) &= \sum_{i\in[n]\setminus j} u_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \\ &= \frac{s_j}{\beta}\cdot(P(s_{k+1},c_{k+1}) + \varepsilon'\cdot(1-\alpha)). \end{aligned}$$

With this new strategy their utility will be

$$\begin{aligned} & u_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \\ &= \sum_{i\in[n]\setminus\{l,l',j\}} [u_{j,i}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})] \\ &+ u_{j,l}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) + u_{j,l'}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \\ &= \sum_{i\in[n]\setminus\{l,l',j\}} [(1-m^*_i)\cdot\frac{a_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) - s_{j_i}}{\beta}\cdot P(s_i,c_i)] \\ &+ (1-m^*_l)\cdot P(s_l,c_l)\cdot\frac{a'_{j,l} + a_{j,l}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)})}{\beta + a'_{j,l}} + \\ &(1-m^*_{l'})\cdot P(s_{l'},c_{l'})\cdot\frac{a'_{j,l'} + a_{j,l'}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)})}{\beta + a'_{j,l'}} \\ &< u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}. \end{aligned}$$

So they cannot increase their utility by choosing this strategy. Note that the ranking of the pools also does not change in this case, because the desirability does not depend on pool size and the number of the active pools remains the same.

3. Consider the case where j removes part of their stake from some pools and leaves it unallocated or allocates it to inactive pools: The utility in this case cannot increase, because as pool member, their utility is always greater than zero.

4. Finally, consider the case where j does a combination of the two above strategies. Then we can prove that their utility will decrease using the inequalities from the previous cases.

□

Lemma 14. *For every* ε'*:* $0 < \varepsilon' < \min\{\varepsilon, P(s_{k+1}, c_{k+1}), \varepsilon_1\}$ *and for every* α*:* $\frac{s_{k+1}}{\beta} < \alpha < 1$ *it holds: In an inner game determined by* $(\vec{m}^*(\varepsilon', \alpha), \vec{\lambda}^*)$*, joint strategies* $\vec{S}^{(\vec{m}^*(\varepsilon', \alpha), \vec{\lambda}^*)}$ *that do not form* k *active saturated pools, whose pool leaders belong to* G*, are not a Nash equilibrium.*

Proof. Let ε': $0 < \varepsilon' < \min\{\varepsilon, P(s_{k+1}, c_{k+1}, \varepsilon_1)\}$ and α: $\frac{s_{k+1}}{\beta} < \alpha < 1$. For simplicity we write $\vec{m}^*$ instead of $\vec{m}^*(\varepsilon', \alpha)$. First, we prove that there is no Nash equilibrium joint strategy $S^{(\vec{m}^*, \vec{\lambda}^*)}$ for which there exist one or more players in G that have not chosen to activate their own pools. Second, we prove that there no Nash equilibrium joint strategy $S^{(\vec{m}^*, \vec{\lambda}^*)}$ for which there exist one or more players $\notin G$ that have activated their pools. Last, we prove that there no Nash equilibrium joint strategy $S^{(\vec{m}^*, \vec{\lambda}^*)}$ for which there exist one or more players who have allocated some of their stake to a pool with total stake more than β or for which there exists some stake that is unallocated or has been allocated to an inactive pool. In more detail

- consider a joint strategy $S^{(\vec{m}^*, \vec{\lambda}^*)}$ for which there exists at least one player j in G that has not chosen to activate their own pool. We will prove that it cannot be a Nash equilibrium, because j will increase their utility if they remove their stake from the other pools and activates their own pool. Note that if j activates their own pool π_j, then π_j will belong to the k most desirable pools, because according to the margins that are specified by the outer game, no pool activated by player $\notin G$ has the same or higher desirability than π_j. This is proved in the second case of the proof of Lemma 13. In addition to this, all the other pools activated by players in G have the same desirability as π_j, not higher. Regarding player j's current utility, by Lemma 12 we have:

$$\begin{aligned} u_j(\vec{S}^{(\vec{m}^*, \vec{\lambda}^*)}) &= \sum_{i \in [n] \setminus \{j\}} u_{j,i}(\vec{S}^{(\vec{m}^*, \vec{\lambda}^*)}) \\ &\leq \frac{s_j}{\beta} \cdot (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1 - \alpha)) \end{aligned}$$

Note that $D_{\max} \leq P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)$ in this inner game, because even if all pools are activated, the pools of players in G have the highest desirability, which is $P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)$, as we proved in the second case of proof of Lemma 13.

If j activates their own pool, then their utility will become:

$$\begin{aligned} u_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) &\overset{\lambda^*_i = s_i}{=} u_{j,j}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \\ &= (m^*_j + (1-m^*_j) \cdot \frac{s_j}{\beta}) \cdot P(s_j, c_j) \\ &\overset{P(s_j,c_j), m^*_j > 0}{>} (1-m^*_j) \cdot \frac{s_j}{\beta} \cdot P(s_j, c_j) \\ &= u_j(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) > 0 \end{aligned}$$

- consider a joint strategy $S^{(\vec{m}^*,\vec{\lambda}^*)}$ that is Nash equilibrium. We will prove by contradiction that no player $\notin G$ has activated their own pool. Specifically we will prove that in such a case, j will increase their utility if they dissolve their pool and allocates their stake to other pools. Therefore this joint strategy cannot be an equilibrium. By the previous case we know that in a joint strategy that is an equilibrium, all players in G have activated their own pools and that these pools have strictly higher desirability than π_j. So π_j does not belong to the k most desirable pools. The utility of j according to $S^{(\vec{m}^*,\vec{\lambda}^*)}$ will be

$$\begin{aligned} &u_j(S^{(\vec{m}^*,\vec{\lambda}^*)}) \overset{\lambda^*_j = s_j}{=} u_{j,j}(S^{(\vec{m}^*,\vec{\lambda}^*)}) \\ &< \frac{s_j}{\beta} \cdot (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)). \end{aligned}$$

This is computed in the same way as in the second case of the proof of Lemma 13. If they choose a different strategy $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$, where they dissolve their own pool and allocates stake to pools whose pool leaders belong to G, so that no pool has stake has more than β, their utility will become:

$$
\begin{aligned}
u_j(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) &= \sum_{i\in[n]\setminus j} u_{j,i}(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \\
&= \sum_{i\in[n]\setminus j} [(1-m_i^*)\cdot \frac{a_{j,i}(S_j^{(\vec{m}^*,\vec{\lambda}^*)}, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j})}{\beta} \cdot P(s_i,c_i)] \\
&= \frac{s_j}{\beta}\cdot(P(s_{k+1},c_{k+1}) + \varepsilon'\cdot(1-\alpha)) \\
&> u_j(S^{(\vec{m}^*,\vec{\lambda}^*)}).
\end{aligned}
$$

- let us suppose that according to joint strategy $S^{(\vec{m}^*,\vec{\lambda}^*)}$), there exists at least one pool π_l with stake more than β. We will prove that in this case $S^{(\vec{m}^*,\vec{\lambda}^*)}$ is not a Nash equilibrium. In more detail, we will prove that a player j that is pool member of this pool π_l can increase their utility by choosing a different strategy $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$. This strategy will be to remove some of their stake, let us say $s_{j_1} < \min\{\beta-\sigma_i, \sigma_l-\beta\}$,

  and to allocate it to an unsaturated pool π_i, whose pool leader belongs to G. Firstly, π_l's pool leader belongs to G, because in an equilibrium, as we proved in the previous two cases, only these players activate their own pools. Recall that these pools have positive desirability. Thus $\text{rank}_l(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \le k$. In addition, we know that there exists an unsaturated pool π_i, whose pool leader belongs to G, because π_l has stake more than β and there exist k pools with pool leaders belonging to G. Moreover $\text{rank}_i(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) \le k$ for the same reason as for π_l. Furthermore, desirability does not depend on pool size and thus:

  $$\text{rank}_l(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \le k$$

  and

  $$\text{rank}_i(S'^{(\vec{m}^*,\vec{\lambda}^*)}_j, \vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}_{-j}) \le k.$$

  Regarding the current utility of j we have:

  $$
  \begin{aligned}
  u_{j,l}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) &= (P(s_{k+1},c_{k+1}) + \varepsilon'\cdot(1-\alpha))\cdot\frac{a_{j,l}(S^{(\vec{m}^*,\vec{\lambda}^*)})}{\sigma_l}, \\
  u_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) &= (P(s_{k+1},c_{k+1}) + \varepsilon'\cdot(1-\alpha))\cdot\frac{a_{j,i}(S^{(\vec{m}^*,\vec{\lambda}^*)})}{\beta}.
  \end{aligned}
  $$

If player j chooses a different strategy $S'^{(\vec{m}^*,\vec{\lambda}^*)}_j$), where they remove s_{j_1} from π_l and allocates to π_i, then only $u_{j,l}$ and $u_{j,i}$ will change. This happens because j's allocations to pools other than π_i and π_l will remain unaffected, and thus the rewards of the other pools will remain unaffected as well. In addition to this, ranking is not affected as we again have k active pools. So

$$u_{j,l}(\vec{S'}_j^{(\vec{m}^*,\vec{\lambda}^*)}, \vec{S}_{-j}^{(\vec{m}^*,\vec{\lambda}^*)}) =$$
$$(P(s_{k+1},c_{k+1}) + \varepsilon' \cdot (1-\alpha)) \cdot \frac{a_{j,l}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) - s_{j_1}}{\sigma_l - s_{j_1}}$$

and

$$u_{j,i}(\vec{S'}_j^{(\vec{m}^*,\vec{\lambda}^*)}, \vec{S}_{-j}^{(\vec{m}^*,\vec{\lambda}^*)}) =$$
$$(P(s_{k+1},c_{k+1}) + \varepsilon' \cdot (1-\alpha)) \cdot \frac{a_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) + s_{j_1}}{\beta}.$$

So

$$u_{j,l}(\vec{S'}_j^{(\vec{m}^*,\vec{\lambda}^*)}, \vec{S}_{-j}^{(\vec{m}^*,\vec{\lambda}^*)}) + u_{j,i}(\vec{S'}_j^{(\vec{m}^*,\vec{\lambda}^*)}, \vec{S}_{-j}^{(\vec{m}^*,\vec{\lambda}^*)}) >$$
$$u_{j,l}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}) + u_{j,i}(\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}).$$

As a result, joint strategy $\vec{S}^{(\vec{m}^*,\vec{\lambda}^*)}$ as described above is not a Nash equilibrium.

- a joint strategy where a player has unallocated stake or stake allocated to an inactive pool is not a Nash equilibrium. This follows because the player can allocate their stake to a pool activated by a player in G and obtain positive utility.

□

5.12.3.5 Equilibria of the Outer Game

In the general case (for any α) the pools that are formed are the ones with the highest $P(s,c)$ (if $\alpha = 0$ this achieves an optimum in terms of social welfare since it minimises the costs of running the system).

Theorem 15. *For every ε': $0 < \varepsilon' < \min\{\varepsilon, P(s_{k+1},c_{k+1}), \varepsilon_1\}$ and for every α: $\frac{s_{k+1}}{\beta} < \alpha < 1$ it holds: Joint strategy $(\vec{m}^*(\varepsilon',\alpha), \vec{\lambda}^*)$ is an ε'-non-myopic Nash equilibrium of the outer game.*

Proof. Let ε': $0 < \varepsilon' < \min\{\varepsilon, P(s_{k+1}, c_{k+1}), \varepsilon_1\}$ and α: $\frac{s_{k+1}}{\beta} < \alpha < 1$. For simplicity we write $\vec{m}^*$ instead of $\vec{m}^*(\varepsilon', \alpha)$.

We will prove that $\forall i \in [n]$ and $\forall (m_i, \lambda_i) \neq (m_i^*, \lambda_i^*)$, it holds that

$$u_i^{\text{outer,up}}(m_i, \vec{m}_{-i}^*, \lambda_i, \vec{\lambda}_{-i}^*) \leq u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*) + \varepsilon'.$$

Specifically, we will examine the following cases for $i \in G$:

- (m_i, λ_i), where $1 \geq m_i > m_i^*$ and $0 < \lambda_i \leq s_i = \lambda_i^*$.
- (m_i, λ_i), where $0 \leq m_i < m_i^*$ and $0 < \lambda_i \leq s_i = \lambda_i^*$.
- (m_i, λ_i), where $m_i = m_i^*$ and $0 < \lambda_i < s_i = \lambda_i^*$.
- $(m_i, \lambda_i) = (m_i, 0)$.

For $i \notin G$ we will examine the case $(m_i, \lambda_i) \neq (m_i^*, \lambda_i^*)$.

These are all the ways in which players can change the strategy described in the theorem. For each case we will prove that in the inner game that is determined by $(m_i, \vec{m}_{-i}^*, \lambda_i, \vec{\lambda}_{-i}^*)$, there is no equilibrium in which the utility of i is higher than $u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*) + \varepsilon'$.

The cases are described below.

- First we will prove that no player $i \in G$ has incentives to increase their margin more than m_i^* in the outer game, regardless of $\lambda_i > 0$. By Lemmas 13,14 the only equilibria of the inner game determined by $(\vec{m}*, \vec{\lambda}*)$ are that ones where k saturated, active pools have been activated by players in G. So for every $i \in G$ we have:

$$\begin{aligned}
&u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*) \\
&= u_i^{\text{outer,up}}(\vec{m}^*, \vec{\lambda}^*) \\
&= (m_i^* + (1 - m_i^*) \cdot \frac{s_i}{\beta}) \cdot P(s_i, c_i) \\
&> \frac{s_i}{\beta} \cdot (1 - m_i^*) \cdot P(s_i, c_i) \\
&\overset{i \in G}{=} \frac{s_i - \lambda_i}{\beta} \cdot (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1 - \alpha)) \\
&+ \frac{\lambda_i}{\beta} \cdot (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1 - \alpha)),
\end{aligned}$$

  where $0 < \lambda_i \leq s_i$.

1. If an $i \in G$ increases their margin by choosing m_i with

$$\frac{P(s_i,c_i) - P(s_{k+1},c_{k+1}) + \varepsilon' \cdot \alpha}{P(s_i,c_i)} \geq m_i > m_i^*$$

and chooses an arbitrary λ_i with $0 < \lambda_i \leq s_i$, then:

$$\begin{aligned}
&u_i^{\text{outer,up}}(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i}) \\
&\leq \max\{(m_i + (1-m_i) \cdot \frac{\lambda_i}{\beta}) \cdot P(s_i,c_i) \\
&+ \frac{s_i - \lambda_i}{\beta} \cdot (P(s_{k+1},c_{k+1}) \\
&+ \varepsilon' \cdot (1-\alpha)), \frac{s_i}{\beta} \cdot (P(s_{k+1},c_{k+1}) + \varepsilon' \cdot (1-\alpha))\} \\
&\leq m_i^* \cdot P(s_i,c_i) + \varepsilon' + (1-m_i^*) \cdot \frac{s_i}{\beta} \cdot P(s_i,c_i) \\
&\leq u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*) + \varepsilon'.
\end{aligned}$$

Note that in the best case there is an equilibrium where (i) i has not activated their own pool, and their utility is at most $\frac{s_i}{\beta} \cdot (P(s_{k+1},c_{k+1}) + \varepsilon' \cdot (1-\alpha))$ by Lemma 12 or (ii) i has activated their own pool, that belongs to the k most desirable pools, and has allocated the remaining stake to the active pools with the highest desirability, which are the pools of players in G. The utility that these pools will give them will be at most $\frac{s_i - \lambda_i}{\beta} \cdot (P(s_{k+1},c_{k+1}) + \varepsilon' \cdot (1-\alpha))$ by Lemma 12.

If there is no equilibrium, recall that

$$u_i^{\text{outer,up}}(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i}) = -\infty$$

.

2. If an $i \in G$ increases their margin by choosing m_i with

$$1 \geq m_i > \frac{P(s_i,c_i) - P(s_{k+1},c_{k+1}) + \varepsilon' \cdot \alpha}{P(s_i,c_i)}$$

and chooses an arbitrary $\lambda_i > 0$, then we can prove that there is no equilibrium in the inner game determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$, where the non-myopic utility of i will be higher than their current lower utility of the outer game denoted by $u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*)$.

This happens because in the inner game determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$ we can prove that there is no an equilibrium where player $k+1$ and

the other players of G have not activated their own pools (note that the desirability of π_{k+1} and of the pools whose pool leaders belong to G, when they are active, are strictly higher than the desirability of π_i, because $m_i > \frac{P(s_i,c_i)-P(s_{k+1},c_{k+1})+\varepsilon'\cdot\alpha}{P(s_i,c_i)}$).

So in the best case, in the inner game determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$ (i) there is an equilibrium where player i has activated their own pool with rank worse than k and has delegated their remaining stake $(s_i - \lambda_i)$ to pools whose pool leaders belong to G, or (ii) there is an equilibrium where i has not activated their own pool and has delegated their whole stake to pools whose pool leaders belong to G, which have the highest desirability.

We will prove that in both cases i's non-myopic utility will not be higher than their current lower utility of the outer game denoted by $u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*)$.

As a result, $u_i^{\text{outer,up}}(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i}) \leq u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*)$.

In more detail:

– In case (ii), by Lemma 12, their utility is at most

$$\begin{aligned}&(P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)) \cdot \frac{s_i}{\beta} \\ &< u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*).\end{aligned}$$

– In case (i), their utility is

$$\begin{aligned}&[m_i + (1-m_i) \cdot \frac{\lambda_i}{\lambda_i}] \cdot (r(\lambda_i, \lambda_i) - c_i) \\ &+ (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)) \cdot \frac{s_i - \lambda_i}{\beta} \\ &\leq P(\lambda_i, c_i) \cdot (0 + (1-0) \cdot \frac{\lambda_i}{\beta}) + \\ &(P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)) \cdot \frac{s_i - \lambda_i}{\beta} \\ &\overset{P(\lambda_i,c_i) \leq P(s_i,c_i), m_i^* > 0}{<} u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*).\end{aligned}$$

We can prove that *there is no equilibrium in the inner game determined by* $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$*, where the players in* G *other than* i *have not activated their own pools*, in the same way as the first case of the proof of Lemma 14. Note that also in this inner game determined by

$(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$: (i) When these players activate their own pools, then these pools always have rank less or equal to k, regardless which other pools have been activated, and (ii) no other pool has strictly higher desirability and offers these players higher utility as members than they get by running their own pools.

Now we will prove by contradiction that *there is no equilibrium in the inner game defined by* $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$*, where the* $(k+1)$*-st player does not activate their own pool.*

Let us suppose that there is a joint strategy $\vec{S}^{(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})}$ that is an equilibrium of the inner game and for which player $k+1$ has not activated their own pool. Then by Lemma 12 it holds:

$$u_{k+1}(\vec{S}^{(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})}) \leq (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)) \cdot \frac{s_{k+1}}{\beta}.$$

Then if they choose a different strategy $S'^{(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})}_{k+1}$ where they activate their own pool, their utility can be increased:

$$\begin{aligned}
&u_{k+1}(S'^{(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})}_{k+1}, \vec{S}^{(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})}_{-(k+1)}) \\
&= P(s_{k+1}, c_{k+1}) \cdot (1 - \frac{\varepsilon' \cdot \alpha}{P(s_{k+1}, c_{k+1})}) \cdot \frac{s_{k+1}}{\beta} \\
&+ \frac{\varepsilon' \cdot \alpha}{P(s_{k+1}, c_{k+1})} \cdot P(s_{k+1}, c_{k+1}) \\
&= (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1-\alpha)) \cdot \frac{s_{k+1}}{\beta} \\
&- \varepsilon' \cdot \frac{s_{k+1}}{\beta} + \varepsilon' \cdot \alpha \\
&= u_{k+1}(\vec{S}^{(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})}) - \varepsilon' \cdot \frac{s_{k+1}}{\beta} + \varepsilon' \cdot \alpha \\
&\overset{\alpha > \frac{s_{k+1}}{\beta}}{>} u_{k+1}(\vec{S}^{(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})}).
\end{aligned}$$

The intuition behind this is that $k+1$ can activate their own pool, that belongs to the k most desirable pools, and in this way can increase their utility, because of the margin that they will take. Note that the desirability of their pool is worse than the desirability of pools whose pool leaders are players in G, but the difference is small, and thus being pool leader is more profitable for $k+1$ than being member of one of their pools.

Recall that in the inner game determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$, if player $k+1$ activates their own pool, this pool belongs to the k most desirable pools, because $P(s_{k+1}, c_{k+1}) - \varepsilon' \cdot \alpha > P(s_{k+2}, c_{k+2})$, where $P(s_{k+1}, c_{k+1}) - \varepsilon' \cdot \alpha$ is its desirability. Note that the desirability of the other pools $\notin G$ is at most $P(s_{k+2}, c_{k+2})$ and that the desirability of π_i, if it is activated, is also lower than $P(s_{k+1}, c_{k+1}) - \varepsilon' \cdot \alpha$. So there exist at most $k-1$ pools activated with higher desirability than that of the $(k+1)$-st player, which causes player $(k+1)$'s pool to belong to the k most desirable pools when it is activated.

- No player $i \in G$ has an incentive to make their margin smaller than m_i^*, given that their pool already belongs to the best k pools in all the equilibria of the inner game determined by $(\vec{m}^*, \vec{\lambda}^*)$.

  In more detail: If an $i \in G$ decreases their margin by choosing $m_i < m_i^*$ and chooses an arbitrary $\lambda_i \leq s_i$, then in the best case there is an equilibrium of the inner game where π_i will again belong to the k most desirable pools, and as a result:

$$
\begin{aligned}
& u_i^{\text{outer,up}}(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i}) \\
& \leq (m_i + (1 - m_i) \cdot \frac{\lambda_i}{\beta}) \cdot P(s_i, c_i) \\
& + \frac{s_i - \lambda_i}{\beta} \cdot (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1 - \alpha)) \\
& \overset{m_i < m_i^*}{<} (m_i^* + (1 - m_i^*) \cdot \frac{\lambda_i}{\beta}) \cdot P(s_i, c_i) \\
& + \frac{s_i - \lambda_i}{\beta} \cdot (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1 - \alpha)) \\
& \leq u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*).
\end{aligned}
$$

- No player $i \in G$ has incentives to commit less stake to their pool in the outer game, because the other existing pools have the same desirability and will not give it higher utility. In more detail: If a player $i \in G$ chooses margin equal to m_i^*, but $\lambda_i < s_i = \lambda_i^*$, then in the best case their pool will belong to

the k most desirable pools, and using Lemma 12, we will have:

$$\begin{aligned}
&u_i^{\text{outer,up}}(\vec{m}^*, \lambda_i, \vec{\lambda}^*_{-i}) \\
&\leq (m_i^* + (1 - m_i^*) \cdot \frac{\lambda_i}{\beta}) \cdot P(s_i, c_i) \\
&+ \frac{s_i - \lambda_i}{\beta}(P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1 - \alpha)) \\
&\leq u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*).
\end{aligned}$$

- Player $i \in G$ has no incentives to set $\lambda_i = 0$, because using Lemma 12, we can prove that in any equilibrium of the inner game determined by $(m_i, \vec{m}^*_{-i}, 0, \vec{\lambda}^*_{-i})$, where they have not activated their own pool, as $\lambda_i = 0$, their utility for being a pool member of other pools will be lower than $u_i^{\text{outer,low}}(\vec{m}^*, \vec{\lambda}^*)$.

- Player $i \notin G$ has no incentives to choose margin $m_i \neq m_i^*$ or to commit stake λ_i less than s_i, because we can prove in the same way as in the second case of the proof of Lemma 14 that in the inner game determined by $(m_i, \vec{m}^*_{-i}, \lambda_i, \vec{\lambda}^*_{-i})$, there is no equilibrium where they have activated their own pool, so margin and stake committed to their own pool do not have an impact on their utility, which by Lemma 12 will be at most $\frac{s_i}{\beta} \cdot (P(s_{k+1}, c_{k+1}) + \varepsilon' \cdot (1 - \alpha))$. For this proof, it is important that players $\notin G$ cannot lower the desirability of the pools activated by players in G by allocating stake to them strategically, because desirability does not depend on pool size. In addition to that, even if player $k+1$ sets margin zero, the desirability of their pool remains strictly lower than the desirability of the pools of all the players of G.

□

5.13 Deployment Considerations

In this section we overview various deployment considerations of our RSS solution as well as we address specific attacks and deviations against our reward sharing scheme. Specifically we overview the following considerations:

- pools that underperform in general.

- players who play myopically.
- pool leaders that do not declare their costs truthfully.
- players who try to gain advantage by exploiting how wealth may compound over time ("the rich get richer" problem) in a series of iterations of the game.

Regarding deployment, the way in which an RSS is facilitated within a PoS cryptocurrency, e.g., [30, 42, 80, 98], is that the ledger is enhanced to enable special transactions that allow players to create a pool and/or to delegate their stake to a pool and reassign it at will during the course of the execution. Recall that in a PoS cryptocurrency, the protocol is executed by participants who get elected in some way based on the stake they possessed in the ledger; informally every protocol message get signed on behalf of particular coin. This coin is verifiably elected for that particular point of the protocol's execution. In the stake pool setting, the PoS protocol will be executed with the pool leaders representing the pool members whenever the coin of a member is elected for protocol participation. Thus when a player delegates its stake, it delegates the right its coins have for participation in the protocol.

5.13.1 Performing Stake Pools

In our RSS, rewards for a pool are calculated based on the declared stake of the pool leader as well as the stake delegated to that pool. So the reward mechanism does not take into account the performance of the pools for assigning the rewards. This opens the door to the following undesired behavior: a pool leader declares a competitive pool and subsequently does not provide the service that it promised (presumably gaining in terms of the actual cost that system maintenance incurs). This can be addressed in the following ways (we leave for future work the formal treatment of these solutions) :

1. we can calibrate the total rewards R to depend also on the total performance of the system as evidenced in the distributed ledger. For instance, in a PoS blockchain, it is possible to count the number of blocks that were produced in a period of time and were included in the ledger and compare that value to its expectation. In case the actual number of blocks is below expectation we may reduce R accordingly (effectively punishing all pools) and in this

way generating a counter-incentive to deviate from system maintenance according to the protocol.

Punishing all pools in case of underperformance has advantages and disadvantages. The advantage is that it does not make the reward mechanism vulnerable to mining games [49, 79] where pools could attack each other to reduce the performance of their competitors.

The disadvantage is that punishing everyone may be hard to parameterise. If R is very sensitive to performance (which means that the punishment is very harsh) then this is unfair to genuinely performing players (who will be losing rewards due to the ill performance of others). On the other hand if the punishment is not harsh maybe it is insufficient to motivate pools to perform the best they can. Moreover when the utility of the players is based on relative rewards finding the right parameterizetion (if exists one) can be even harder.

2. we can penalise the pool that underperforms. Note that in order to facilitate this the ledger should be able to provide enough evidence if a pool underperforms or not. So in order to facilitate this the underlying blockchain should be also "fair" (in the sense of [108]). Otherwise mining games can make a pool appear as deviant and get penalised unfairly. It is an interesting question to design such robust performance metrics that can be used in the context of a reward sharing scheme.

5.13.2 Players who Play Myopically

As we have already mentioned if pool leaders played myopicly this would lead to unstable configurations with unrealistically high margins that are not competitive. Also pool members have incentives to play non-myopicly because in this way they enhance competition among pools which leads to lower margins. However there is the case that this motivation is not obvious for many pool members, thus we suggest how it is also possible to force pool members to play non-myopically.

The key idea is that the effect of delegation transactions should be considered only in regular intervals (as opposed to be effective immediately) and in a certain restricted fashion. One way to achieve this is by restricting delegation instructions to a specific subset of stakeholders at any given time in the ledger operation and making them effective at some designated future time of the ledger's operation.

Due to these restrictions, players will be forced to think ahead about the play of the other players, i.e., stakeholders will have to play based on their understanding of how other stakeholders will as well as the eventual size of the pools that are declared.

5.13.3 Costs and Incentive Compatibility

In our analysis, we assumed for simplicity that the costs are publicly known. However in reality only the player knows the actual costs for participating in the collaborative project and thus it may lie about it in the cost declaration in order to create a more competitive pool. This will happen when the players may increase their utility if they lie about their cost. This problem is one of mechanism design which has objective to design an incentive compatible mechanism, i.e., a mechanism that gives incentives to players to declare their costs truthfully. We next argue that, in fact, our RSS is incentive-compatible as presented. Let us consider the perfect strategies from Definition 18 in which the utilities are given by Equation 5.3. Suppose that a pool leader j declared a different cost $\hat{c}_j$, but remained pool leader. Since $P(s_j,\hat{c}_j) - P(s_j,c_j) = c_j - \hat{c}_j$, the player will not get any benefit from lying. To see this, let $u_j(\hat{c}_j|c_j)$ denote the utility when the player declares cost $\hat{c}_j$ instead of the true cost c_j. Then by taking into account the cost, we have $u_j(\hat{c}_j|\hat{c}_j) = u_j(\hat{c}_j|c_j) - c_j + \hat{c}_j$. Also from Equation 5.3, we see that $u_j(c_j|c_j) - u_j(\hat{c}_j|\hat{c}_j) = P(s_j,c_j) - P(s_j,\hat{c}_j)$. Putting them together we see that $u_j(\hat{c}_j|c_j) = u_j(c_j|c_j)$, thus the player has no reason to lie. With similar reasoning, a pool leader has no reason to lie by raising its cost so much that the rank of its pool increases above k. Similar considerations, show that no pool member (i.e., a player whose pool, if created, would have had a rank at least $k+1$) has an incentive to lie. This includes the special case of the player with rank $k+1$. As a conclusion, we see that under the assumption that the players end up at the perfect strategies, it is a dominant strategy to declare the true cost.

5.13.4 "Rich Getting Richer" Considerations

When our RSS will be deployed in a proof of stake system, our game will be played in epochs and each iteration will succeed the previous one. A special consideration here is what frequently is referred to as the "rich get richer" problem, i.e., the setting where the richest stakeholder(s) amass over time even more wealth due

to receiving rewards something that leads to an inherently centralised system (it is sometimes believed that this issue is intrinsic to only PoS systems but in fact it equally applies to PoW systems, cf. [74]). We address this issue by observing that the maximum rewards each pool earns at each epoch are in the range $[R/(1+\alpha)k, R/k]$ with $\alpha \in [0, +\infty$. This determines how much more rich pools (i.e., pools with rich pool leaders who can pledge more stake to their pool) can earn. Thus using α we can control the disparity created by the reward mechanism. Observe that if we choose $\alpha \to 0$ we achieve a perfectly "egalitarian" effect because rich pools and poor pools of the same size are receiving exactly the same rewards, something that does not affect the relative stake from epoch to epoch in case we do not take into account margins. This makes a "rich dollar" and a "poor dollar" (cf. [74]) have the same value, but it comes with the downside that it makes our system vulnerable to the Sybil behaviours.

5.14 Tradeoffs

Tradeoff between Sybil resilience and efficiency: recall that in the Nash equilibrium that we describe the pool leaders will be the players with the highest potential profit. If α is zero the pool leaders will be the players with the lowest cost but we will have no Sybil [43] protection. On the other extreme, by selecting a very large α, we can obtain a potentially inefficient solution in which the pool leaders might be the k "wealthiest" but the Sybil resilience of the system improves (the lower bound for the Sybil attack increases).

Tradeoff between Sybil resilience and egalitarianism The lower α we select, the smaller difference in power a "rich and a poor dollar" have but also the lower Sybil resilience we guarantee.

5.15 Related Work

Note that in our case we follow a "representative democracy" approach, broadly followed by [38, 80, 85] where the stakeholders can empower other stakeholders to represent them in project maintenance and subsequently share the rewards. Given that empowering is performed via stake as recorded in the ledger, representatives can be thought to form "*stake pools*" in analogy to the mining pools of Bitcoin.

As we have already referred a number of previous works considered the incentives of mining pools in the setting of proof of work cryptocurrencies (as opposed to proof of stake ones) such as Bitcoin [47, 112, 114, 118]. The main differences between mining pools in Bitcoin and stake pools in our setting are that (i) in Bitcoin all pool members perform mining and hence incur costs, while in PoS setting, only the pool leader runs the underlying protocol and incurs a cost while delegators have no cost, (ii) in Bitcoin each pool leader can choose a different way to reward pool members/miners while in our setting we prescribe a specific way for rewards to be shared between pool members.

Regarding centralization, Arnosti and Weinberg, [11], have established that some level of centralisation takes place in Bitcoin in settings where differences in electricity costs are present between the miners. Also according to [86] in a setting where each unit of resource has a different value depending on the distribution of the resources among the players, miners have incentives to create coalitions. These results are inline with our (even more centralised) negative result on fair RSS's for the PoS setting, cf. Section 5.4.3. Another aspect we do not explore here, is the instability of such protocols when the rewards come mostly from transaction fees; this was explored in [34, 120].

With respect to PoS blockchain systems, a different and notable approach to stake pools is to use the stake as voting power to elect a number of representatives, all of equal power, as in delegated PoS (DPoS) [85]; for example, the cryptocurrency EOS [64] has 21 representatives (called block producers). This type of scheme differs from ours in that (i) the incentives of voters are not taken into account thus issues of low voter participation are not addressed, (ii) elected representatives, despite getting equal power, are rewarded according to votes received; this inconsistency between representation and power may result in a relatively small fraction of stake controlling the system (e.g., at some point, controlling EOS delegates representing just 2.2% of stakeholders was sufficient to halt the system,[5] which ideally could withstand a ratio less than 1/3), (iii) it may leave a large fraction of stakeholders without representation (e.g., in EOS, at some point, only 8% of total stake is represented by the 21 leading delegates[3]).

Yet another alternative to stake pools is that of Casper [30], where players can propose themselves as "validators" committing some of their stake as collateral. The committed stake can be "slashed" in case of a proven protocol deviation.

[3]Statistics extracted from http://eos.dapptools.info/#/block-producers on July 27th, 2018.

This type of scheme differs from ours in that (i) stakeholders wishing to abstain from protocol maintenance operations have no prescribed way of contributing to the mechanism (as in the case of voting in DPoS or joining a stake pool in our setting), (ii) a small fraction of stake may end up controlling the system while at the same time leaving a lot of stake decoupled from the protocol operation; this is because substantial barriers may be imposed in becoming a validator (e.g. see the EIP proposal for Casper[4]). This can make it infeasible for many parties to engage directly; on the other hand reducing this threshold drastically may make the entry barrier too low and hence still allow a small amount of stake to control the system. As a separate point, it is worth noting that for both the above approaches there is no known game theoretic analysis that establishes a similar result to the one presented herein, i.e., that the mechanism can provably lead to a Nash equilibrium with desirable decentralisation characteristics that include a high number of protocol actors and Sybil attack resilience.

The compounding of wealth in PoS cryptocurrencies was studied in [50] where a new notion denoted by "equitability" is introduced to measure how much players can increase their initial fraction of stake. Also they prove that a "geometric" reward function is the best choice for optimizing equitability under certain assumptions; we remark that it is a folklore belief that PoS systems are inherently less equitable than ones based on PoW, however this belief seems to be unfounded, cf. [74]. With respect to equitability we show that by calibrating one of our parameters called *Sybil resilience parameter* to be small our system becomes "equitable" in the sense of providing similar rewards to stake pool leaders independently of their wealth.

From a game-theoretic perspective, our setting has certain similarities to cooperative game theory in which coalitions of players have a value. In our setting the players have weights (stake) and they are allowed to split it into various coalitions (pools). Our objective is to have a given number of equal-weight coalitions, which contrasts with the typical question in cooperative game theory on how the values of the coalitions are distributed (e.g., core or Shapley value) in such a way that the grand coalition is stable [105]. Actually, the games that we study are variants of congestion games with rewards on a network of parallel links, one for every potential pool. The reward on each link is determined by the reward function, which essentially determines an atomic splittable congestion game. But

[4]https://eips.ethereum.org/EIPS/eip-1011.

unlike simple atomic splittable congestion games [103], our games have different reward for pool leaders and for pool members. There are two main research directions for such games: whether they have unique equilibria and how to efficiently compute them [24]. Regarding the question of unique inner equilibria the most relevant paper to our inner game is [104] (but see also [23, 110]) which shows that under general continuity and convexity assumptions, games on parallel links have unique equilibria. However, the conditions on convexity do not meet our design objectives and they do not seem to be useful in our setting.

Our work is related to two aspects of delegation games, which are games that address the benefits and other strategic considerations for players delegating to someone else to play a game on their behalf, such as owners of firms hiring CEO's to run a company. The first aspect is somewhat superficially related to this work in pool formation the pool members delegate their power to pool leaders. The second aspect which is much more relevant to our approach is that delegation changes the utility of the players (for example, by considering "credible threats" [116, 117]) or creates a two-stage game [53, 119, 123]. A typical two-stage delegation game is non-myopic Cournot competition [55] in which in the outer game the firms (players) decide whether to be profit-maximizers or revenue-maximizers, while in the inner game they play a simple Cournot competition [96]. Unlike our case, the inner Cournot competition has a simple unique equilibrium which defines a simple two-stage game.

Another research area that is relevant to this work is mechanism design, because participants may have an incentive not to reveal their true parameters, e.g., the cost for running a pool [103, 124].

In the proof of work setting, [56] considers reward sharing rules for proof of work systems under the assumption of discounted expected utility and identifies schemes that achieve fairness. Furthermore, an axiomatic approach to reward schemes of proof of work systems is taken in [36] in order to study fairness, symmetry, budget balancing and other properties. Unlike our work that considers *incentives for pool formation* with desirable properties, these two papers study intrinsic properties of the system *given* an existing pool formation.

Finally another work, [83] that studied a parameterized notion of decentralization, where, in an ideal system, all participants should exert the *same power* in running the system, independently of their stake. This is a significantly more demanding notion of decentralization than the one considered here, where in an

ideal system, participants *exert power proportional to their stake*. It is argued in [83] that in order for a system to achieve full decentralization, there must exist a strictly positive Sybil cost, that is, the cost of running two or more nodes should be higher when the nodes belong to the same entity than to multiple entities. Clearly in systems with anonymous users, Sybil costs cannot be positive and such concept of decentralization is impossible.

Chapter 6

Anti-censorship Mechanism

6.1 Introduction and Related Work

As we have mentioned in the previous chapter in order for our RSS to get deployed in a PoS cryptocurrency special transactions should be enabled. To be specific, the following special transactions should be enabled: (i) a *pool registration transaction* (or *pool registration certificate*) with which a player declares that it wishes to create a pool with specific cost, margin and pledged stake (ii) a *stake delegation transaction*(or *stake delegation certificate*) with which a player declares that it wishes to delegate an amount of stake to an existing pool.

In a nutshell, when our RSS is deployed in a PoS cryptocurrency, we consider that a pool is registered or a delegation takes place when the relevant transactions are added in the ledger. This approach is followed by many PoS cryptocurrencies. Some examples are Cardano's stake pools [3] and Ethereum-2.0's "validators"[1]. This means that some of the already registered pools should include them in the block they will sign when they are elected. But do they have incentives to do so?

In the honest/malicious setting when a PoS blockchain protocol can guarantee liveness [80], then we know that under certain assumptions all the transactions will be eventually added to the ledger. Observe that the honest players include transactions in their blocks regardless their content (if they are valid). Thus no transaction can be censored for a long period of time regardless its content, if the majority of the players are honest.

However in a rational setting where all the players are rational it is not obvious that a transaction will eventually be included in the ledger.

[1]Please refer to `https://github.com/ethereum/eth2.0-specs`.

For example imagine a scenario where a pool registration transaction is issued but none of the existing pool leaders has incentives to include it in its block, because if it does so and allows the new pool to be created, the system will stabilize in the following state: the existing pool leaders continue to have saturated pools, but they have decreased their margin in order to be more competitive than the new pool that has been recently created.

6.1.1 Our Contribution and Roadmap

In this chapter we design an anti-censor mechanism that incentivizes at least one pool leader to include the above transactions (Section 6.2). Furthermore, we model the problem as various games and we present and prove which are the Nash equilibria that appear when this mechanism is used [2] (Section 6.3). In Section 6.1.2 we refer to related works.

6.1.2 Other related work

We remark that the central game-theoretic problem faced by pool members that prefer the formation of a new pool is how to coordinate. Coordination of players to select an appropriate equilibrium is a vast topic in game theory. More relevant to our approach is the investigation of effects of "cheap talk" (see for example [13, 40, 51, 52]) in coordination games [39, 122].

6.2 Our Anti-censorship Mechanism

Our anti-censor mechanism consists of two parts:

1. the first part takes advantage of the ability of the distributed ledger to produce unpredictable randomness in regular time intervals called *rounds* [27].

2. the second part introduces compounding stake delegation transactions with pool registration transactions that will be possibly censored.

[2] these results are based on our under submission paper "Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. Incentives Against Power Grabs or How to Engineer the Revolution in a Pooled Proof of Stake System, 2020".

In each round we assume that the ledger produces an unpredictable number ρ used to select uniformly and independently a random stake pool. This pool will be dissolved as a form of "randomized audit". Their members will be invited to join it again and thus they should be online. Moreover, the stake delegation transactions that the members should send in order to join their pool again should be compounded with a pool registration certificate.

This certificate is supposed (but not restricted) to refer to a pool π_j that will be possibly censored by the existing pools. In this case if the pool leader wishes to censor π_j, then it has also to censor the stake delegation certificates that include the related pool registration certificate. If this happens, then the participants who added this certificate are not considered as members of this pool. This could reduce the pool leader's utility (we assume that if the pool has fewer members than previously, the utility of the pool leader will not be higher; this is in line with the previous chapter where a system that uses our RSS stabilizes in a state with k pools with size $1/k$, which means that if a pool loses some members, then its size will become smaller than $1/k$ and its rewards will be reduced).

On the other side, each member has the dilemma between compounding its stake delegation transaction with a pool π_j that pool leaders would like to censor or with an already existing pool. If it compounds its stake delegation transaction with π_j then there is the possibility that the pool leader rejects its transaction in order to censor π_j, which means that it will lose its rewards for this round, because its stake will remain undelegated. On the other hand, if the pool leader does not reject its transaction then the new pool will be registered and thus the utility of the member may increase in the future. Observe that its utility may increase in the future, because (i) it will have an extra option, when it chooses to which pool it will delegate its stake and also (ii) the margin of the existing pools may decrease due to competition.

6.3 Game Theoretic Analysis of our Mechanism

6.3.1 Single Round-Single Pool Game

Initially we focus on a single pool (the pool that is selected to get dissolved) and a single round and we consider a game which is a simplified case of the game-theoretic situation. Observe that examining a single pool and a single round is

a simplified case because: (i) the utilities of both the pool leader and the pool members depend on many other external factors such as the behavior of the other pools (ii) the game is played repeatedly in a potentially dynamic environment.

The game is played between a pool leader and the members of the pool that was randomly selected as randomized audit. For simplicity we consider that there exist two available pool registration certificates: the first one refers to a new pool and the second one to an existing pool.

The pool leader has two possible strategies:

1. censor which means that it rejects all the stake delegation transactions that register the competitive new pool. Note that it does not make sense for the pool leader to reject some of the transactions and not all because : if just one stake delegation transaction compounded with the registration transaction of the new pool is accepted, then the new pool will be created and thus there is no reason for the pool leader to censor the other members and lose rewards.

2. notcensor which means that it accepts all the stake delegation transactions that register the competitive new pool.

On the other hand, each pool member has two strategies:

1. capitulate, where it capitulates and compounds its stake delegation transaction with the pool registration certificate of the existing pool

2. rebel, where it "rebels" against the cartel and compounds its stake delegation transaction with the pool registration certificate of the new pool.

Observe that one possible reason for a member to choose capitulate is that it is afraid of its transaction getting rejected and thus its stake remaining undelegated if it chooses rebel. This could happen if very few members with very little stake chose rebel and the pool leader preferred to lose these members in order to censor the new pool.

Based on the above reasoning, we define the strategies and the utilities of the players and we also find all the pure Nash equilibria of this game.

Strategies of the players: the set of strategies for the pool leader is $\{\mathsf{censor}, \mathsf{notcensor}\}$ and the set of strategies for the pool members is $\{\mathsf{capitulate}, \mathsf{rebel}\}$. The strategy

of the pool leader will be denoted by S_P and the strategy of a pool member i will be denoted by S_i.

Utilities: recall that a utility function takes as input the joint strategy (a vector with the strategies of all the players) and outputs the quantity that the players want to maximize such as profit. Suppose that x is the stake of pool members that select capitulate. The utility of the pool leader is:

- u_x^P, when either it chooses censor or it chooses notcensor and all members choose capitulate.
- u'^P when it chooses notcensor and at least one member chooses rebel. This case captures the setting when the new pool registration is installed and the stake-pool line up of the system is modified.

The utility of a pool member i is :

- u_x^i when the pool leader chooses censor and i chooses capitulate. This case gives the utility that member i obtains by being a member of a pool that has total delegated stake x.
- u_i' when the pool leader selects notcensor and at least one player chooses rebel. This case captures the setting when a new pool is registered and a pool member can benefit from the new stake-pool lineup.

Let n be the number of players, s_i the stake of member i, s_P the stake of the pool leader, x the stake of pool members that select capitulate and $y = s_1 + ... + s_n + s_P$. Using the above notation we define the utility of the pool leader and the pool members in our game as follows: The utility of the pool leader U_P in a joint strategy $(S_1, S_2, ..., S_n, S_P)$ will be

$$U_P(S_1, S_2, ..., S_n, S_P) = \begin{cases} u_x^P \text{ if } S_P = \mathsf{censor} \\ u'^P \text{ if } S_P = \mathsf{notcensor} \wedge \exists i\, S_i = \mathsf{rebel} \\ u_y^P \text{ if } S_P = \mathsf{notcensor} \wedge \forall i\, S_i = \mathsf{capitulate} \end{cases}$$

The utility of the pool member i denoted by U_i in a joint strategy $(S_1, S_2, ..., S_n, S_P)$

will be

$$U_i(S_1,S_2,...,S_n,S_P) = \begin{cases} 0 \text{ if } S_P = \mathsf{censor} \wedge S_i = \mathsf{rebel} \\ u_x^i \text{ if } S_P = \mathsf{censor} \wedge S_i = \mathsf{capitulate} \\ u_i' \text{ if } S_P = \mathsf{notcensor} \wedge \exists i\, S_i = \mathsf{rebel} \\ u_y^i \text{ if } S_P = \mathsf{notcensor} \wedge \forall i\, S_i = \mathsf{capitulate} \end{cases}$$

Observe that when the pool leader chooses "notcensor" and all the players capitulate, the outcome is that, independently of the presence of the new pool, pool members will remain with the old pool and hence the utility of a player i will be u_y^i not u_i'.

Pure Nash equilibria of the game: the following theorem characterizes the pure Nash equilibria of this game. It states that there are two types of equilibria: a unique equlibrium in which censor is successful and every member capitulates, and a class of equilibria in which members with sufficient stake can manage to force the pool leader not to censor.

Theorem 16. *Let F be the following event:*

$\big((\textit{there exists unique } i \textit{ such that } S_i = \mathsf{rebel}) \cap ((S_i = \mathsf{rebel}) \Rightarrow (u_i' \geq u_y^i))\big)$

and J the event that there exist more than one players with rebel*. Assuming that (i) $\forall i$ it holds $u_x^i > 0$ where $x \geq s_i + s_P$ and (ii) there exists a player i for whom it holds $u_i' > u_y^i$ all the pure Nash equilibria of our game are the following:*

- $A = (S_1,...,S_n,\mathsf{notcensor})$ *such that*

$$\left(u'^P \geq u_a^P \textit{ where } a = \sum_{j:S_j=\mathsf{capitulate}} s_j\right) \cap (J \cup F)$$

- $B = (S_1,...,S_n,\mathsf{censor})$ *such that* $(\forall i\ S_i = \mathsf{capitulate})$.

Note that we used assumption (i) in order to prove uniqueness of equilibria B and (ii) in order to prove uniqueness of equilibria A.

Proof. First we will prove that joint strategy A is an equilibrium.

1. If the pool leader chooses censor then their utility will become u_a^P where $a = \sum_{j:S_j=\mathsf{capitulate}} s_j$ which is at most their current utility that is u'^P given that there exists at least one player with rebel.

2. Regarding the other players:

- If F holds : If the player i who has chosen rebel changes their strategy to capitulate then their utility will become u_y^i which is at most his current utility which is u_i'. If a player j with $S_j = \text{capitulate}$ chooses rebel then their utility will not change because already one player has chosen rebel and they have not been censored.
- If J holds: If a player j with $S_j = \text{capitulate}$ chooses rebel strategy then their utility will not change. The same happens with a player who has chosen rebel.

Now we will prove by contradiction that there is no other equilibrium than A where the pool leader has chosen notcensor.

- Let us assume that there is an equilibrium where the pool leader has chosen notcensor but the condition ($u'^P \geq u_a^P$ where $a = \sum_{j:S_j=\text{capitulate}} s_j$) does not hold. Then the pool leader can increase their utility from u'^P to u_a^P by changing his strategy to censor. Note that the utility of the pool leader will be u'^P because we can prove by contradiction that there is no equilibrium where all players have chosen capitulate and the pool leader notcensor. Specifically if there was such an equilibrium then the player for whom it holds $u_i' > u_y^i$ could increase their utility by choosing rebel.
- Let us assume that that there is an equilibrium where the pool leader has chosen notcensor but the condition ($F \cup J$) does not hold. This means that either (a) there exist a unique player with rebel strategy but for this player it holds $u_i' < u_y^i$ or (b) all players have chosen capitulate. In case (a) the player with rebel can increase their current utility u_i' to u_y^i if they choose capitulate and in case (b) the player for whom it holds $u_i' > u_y^i$ can increase their utility by choosing rebel.

Now we will prove that joint strategy B is an equilibrium.

1. If the pool leader changes strategy and chooses notcensor then their utility will remain the same because all the players have chosen capitulate.
2. If a player changes their strategy and chooses rebel then their utility will become zero that is smaller than their current utility u_y^i.

Now we will prove by contradiction that there is no equilibrium where the pool leader has chosen censor and it is different from B. If there was at least one player

with rebel then they could increase their utility from zero to u^i_x, where $x \geq s_i + s_P$ by choosing capitulate. □

6.3.2 Multiple Rounds-Single pool Game with Static Strategies

In this subsection we extend the game to multiple rounds, where in each round the same pool is selected to be dissolved.

Strategies: the players decide in the beginning their strategies for all the rounds. The strategy of each member is to choose capitulate or rebel for each round and the strategy of the pool leader to choose notcensor or censor for each round. We assume that when the pool leader chooses notcensor or a member chooses rebel in one round then in the following rounds they are restricted to do the same. In the following subsection we remove this assumption and we examine also adaptive strategies. Formally

Definition 22. *Let $s_{i,j}$ be the strategy of player $i = 1,\ldots,n$ for round $j = 1,\ldots,m$. Let $\vec{s}^r_i$ be the vector with the strategies of player i until round r (including r) and $\vec{s}_i^{-r}$ be the vector with the strategies of player i after round r. Let $\vec{S}^r = (\vec{s}^r_P, \vec{s}^r_1, ..., \vec{s}^r_n)$ be the vector with the joint strategy until round r and $\vec{S}^{-r} = (\vec{s}_P^{-r}, \vec{s}_1^{-r}, ..., \vec{s}_n^{-r})$ the vector with the joint strategy after round r.*

Utilities: the utility of a player i for a game with m rounds is $u_i = u_{i,1} + ... + u_{i,m}$ where $u_{i,j}$ is the utility of player i in round j. If the pool leader chooses notcensor in a round and there exists a player with rebel, then in all the following rounds the utility of the members will be u'_i and the utility of the pool leader u'^P. Otherwise the utilities of each player during each round are defined as in the one round game we described above.

Our results: given that our game is progressing in multiple rounds we are searching for joint strategies that are more stable than Nash equilibria and thus are Subgame Perfect Nash Equilibria [78]. This means that if we restrict these joint strategies in a number of rounds (a final part of the initial game) they continue to be Nash equilibria in this restricted game. Formally

Definition 23. *A subgame r will be a game as described above beginning from round $r+1$ and ending after round m.*

Definition 24. *A joint strategy $\vec{S}^m$ will be a Subgame Perfect Nash Equilibrium if for every subgame r it holds that $\vec{S}^{-r}$ is a Nash Equilibrium.*

The following theorem describes a class of joint strategies that are Subgame Perfect Nash Equilibria and extensions of the single-round equilibria. In more detail, the class contains all equilibria in which during an initial phase the new pool is censored (i.e., the pool leader chooses censor and all the members capitulate) and at some point the "revolution" succeeds meaning that the new pool is registered (i.e., the pool leader chooses notcensor while some members rebel). Observe that this class of joint strategies that are Subgame Perfect Nash Equilibria include also the extreme cases where the initial phase is empty or covers all rounds.

Theorem 17. *Assume that (i) every member has strictly positive utility when the new pool is censored and the member's stake delegation transaction gets accepted i.e. $u_x^i > 0$, where $x \geq s_i + s_P$, and that (ii) the utility of the pool leader does not increase if the new pool is registered i.e., $u'^P \leq u_y^P$.*

The following joint strategies for the game of m rounds are Subgame Perfect Nash Equilibria for every $l \in \{1,...,m\}$: The pool leader selects $s_{P,j} =$ censor, at every time step $j \leq l$, and $s_{P,j} =$ notcensor for $j > l$. All pool members select $s_{i,j} =$ capitulate for $j \leq l$. At time $j = l+1$, there exist either two or more members that select rebel *or a unique player i that selects* rebel*; in the latter case, player i must prefer the new pool from the current situation, i.e., $u'_i \geq u_y^i$, where y is the total stake of the old pool. Furthermore, the total stake of* rebel *members at time $l+1$ must make the current pool unattractive to the pool leader, i.e., $u'^P \geq u_a^P$ where $a = \sum_{i:s_{i,l+1}=\text{capitulate}} s_i$.*

Proof. We will prove that these joint strategies are Subgame Perfect Nash Equilibria. In subgame $m-1$ (which includes only round m) the joint strategies $\vec{S}^{-(m-1)}$ are Nash equilibria as we have proved in the previous theorem. Now we will take an arbitrary subgame $m-i$ and we will prove that $\vec{S}^{-(m-i)}$ as described above is a Nash equilibrium of this subgame. $\vec{S}^{-(m-i)}$ can be:

1. in all rounds pool leader has chosen notcensor and all the conditions described in the theorem for that case hold.

2. in some rounds in the beginning pool leader has chosen censor and all the members capitulate and from one point and after the pool leader choose notcensor and the relevant conditions hold.

3. in all rounds pool leader has chosen censor and all the members capitulate.

In the first case:

- if the pool leader changes their strategy and chooses censor initially in some rounds (or in all rounds) then (i) their utility for these rounds will not increase because it holds $u'^P \geq u_a^P$ where $a = \sum_{i:s_{i,j}=\text{capitulate}} s_i$ and (ii) their utility in the following rounds will not change because they have chosen notcensor and there exists at least one member with rebel which means that their utility is u'_P regardless what happens in the previous rounds.

- if a pool member chooses rebel in an earlier round then this will not change their utility because already in each round there exist a player with rebel and the pool leader has chosen notcensor. If a pool member i chooses rebel later or not at all, then if there exists other member with rebel in these rounds then their utility will not change. If there is no other member with rebel the utility of i will not increase because $u'_i \geq u^i_y$. Also their utility in the following rounds will be the same.

In the second case:

- if the pool leader chooses notcensor earlier then their utility will not change because all the members have chosen capitulate. If the pool leader chooses censor later then their utility will not increase because of the same arguments as above. if a member i choose rebel earlier then their utility will not increase given that the pool leader has chosen censor and it holds $u^i_y \geq 0$.

- if a member chooses rebel later then if there exists other player with rebel then their utility will not change. If there are not any other players with rebel then their utility in these rounds will not increase because $u'_i \geq u^i_y$, and their utility in the next rounds will be the same (because there exist a player with rebel and the pool leader has chosen notcensor).

In the third case:

- if the pool leader chooses notcensor then their utility will not change because there is no member with rebel.
- if a member choose rebel then their utility will not increase because $u_y^i \geq 0$.

□

Note that if assumption (ii) does not hold, it means that the pool leader will increase its utility, if it lets the new pool get registered. A scenario where this happens is the following: the new pool has so good characteristics (low margin, low cost,high pledged stake) that it is more profitable for the pool leader of the pool we examine in the game to dissolve its pool and join the new pool. Thus we do not examine this case because the pool leader does not have incentives to censor the new pool.

6.3.3 Multiple Rounds-Multiple Pools Game with Adaptive Strategies

In this subsection we turn our attention to a more realistic and interesting setting in which the players adapt their strategies based on what happened in the previous rounds. In this setting we show a richer set of joint startegies that are Nash equilibria where a few members of a pool π_i rebel in a round against the cartel if in a previous round a player of a pool π_j has rebelled but was censored and lost its income. This means that the member of pool π_j who rebelled and got censored in the previous round gave a signal (an information diffused in the peer to peer network) to the members of pool π_i.

Initially for simplicity we examine a game where π_i and π_j are the same pool. After that we generalise the game and we allow pools π_i and π_j to be different and represent the pools that have been selected as randomized audits. This is meaningful because the signal from a player can be sent to members of other pools and in this way the revolution can be facilitated in a smaller period compared to waiting until the same pool is elected as randomised audit.

This is inline with a PoS system operation where transactions are assumed to be communicated in an underlying peer to peer network and are thus accessible to all players despite the fact that the players that operate the current set of stake pools may choose to censor them and postpone their inclusion into the ledger state.

Observe that we use the concept of "signal" and we do not consider a scenario where the members of a pool provide a commitment that they will select the rebel strategy in order to "force" the pool leader to select notcensor. The reason is that such a commitment is impossible to be enforced off-chain and if it is facilitated on chain the pool leaders would censor these commitments.

Thus members have to rely on "cheap-talk"[13, 40, 51, 52] (with no direct effect on utility) or signals (with some direct effect on utility) to coordinate. A signal that a member can give to others is to select rebel and forfeit the income for one round. When this happens and the pool leader censors it, the player's choice does not appear on-chain. Still, other players learn about the suppressed strategy from off-chain communication and by checking that it did not appear on-chain.

6.3.3.1 Two Rounds-Single Pool Game with Adaptive Strategies

At this point we examine a game that consists of two rounds and in both rounds the same pool is examined. After that we examine a two round game where each round refers to a different pool.

Strategies: we extend the strategies of the previous subsection to allow for communication and signaling. Now the strategy of the second round can depend on the strategies selected in the first round.

The strategy of a member i is $\vec{s}_i = (s_{i,1}, s_{i,2}(s_{1,1}, ..., s_{n,1}, s_{P,1}))$. The strategy of the pool leader is $\vec{s}_P = (s_{P,1}, s_{P,2}(s_{1,1}, ..., s_{n,1}))$. We now define a natural set of strategies X that allow the members to coordinate in the second round by responding to choices of the first round.

- Let X be the strategy of pool members in which they rebel in the second round when they see a member with rebel in the first round, i.e., $X =$ rebel if there exists i such that $s_{i,1} =$ rebel, and capitulate otherwise.
- Let Y be the strategy of the pool leader in which they select notcensor in the second round, when they see a member with rebel in the first round, i.e., $Y =$ notcensor if there exists i such that $s_{i,1} =$ rebel, and censor otherwise.

Theorem 18. *Assuming that for all i, u_x^P and u_x^i is increasing in x and for all i and x, $u_x^i \geq 0$, then the following joint strategies are Nash equilibria:*

- *the strategy of the pool leader is* $s_P = (censor,\ Y)$*, when* $u'^P \geq u^P_{y-a}$*, where* y *is the total stake of pool members (including pool leader) and* $a = \sum_{i:s_{i,2}=X} s_i$ *is the stake of pool members that select strategy* X*.*

- *the strategy of the pool members satisfy* $L \cap J \cap F$*, where* J *is the event that* $\exists i, j$ *such that* $s_{i,2} = s_{j,2} = X$*,* L *is the event that there exists unique* i *such that* $s_{i,1} =$ *rebel and* F *the event that* $s_{i,1} = rebel \Rightarrow (u'^P \leq u^P_{y-s_i}) \cap (u'_i \geq 2u^i_y)$*.*

Proof. Let i be the member that chooses rebel in the first round.

- If the pool leader changes their strategy and chooses (notcensor, notcensor) or (notcensor,Y_1) where Y_1 is an arbitrary function that is equal to notcensor for the given strategies of the first round, then the strategy of the pool leader will become $2 \cdot u'^P$ that is not higher than their current utility that is equal to $u^P_{y-s_i} + u'^P$ (because of the first part of F event).

  If the pool leader chooses (notcensor, censor) or (notcensor,Y_2) where Y_2 is an arbitrary function that is equal to censor for the given strategies of the first round then their utility will become again $2 \cdot u'^P$ that is not higher than their current utility.

  If the pool leader chooses (censor, censor) or (censor,Y_3)
  where Y_3 is an arbitrary function that is equal to censor for the given strategies of the first round then their utility will become at most $u^P_{y-s_i} + u^P_{y-\alpha}$ that is not higher than their current utility.

  If the pool leader chooses (censor,Y_4) where Y_4 is an arbitrary function that is equal to notcensor for the given strategies of the first round then their utility will remain the same.

- If player i (for whom it holds $s_{i,1} = rebel$) chooses (rebel, M) where M is an arbitrary strategy then their utility will remain the same, because there exist at least two members that in the second round have chosen X strategy (event J). Note that X in this case is equal to "rebel" because $s_{i,1} =$ rebel and Y equal to notcensor.

  If player i chooses capitulate in the first round (regardless what they choose in the second round) then their utility will become at most $2 \cdot u^i_y$ not higher than u'_i which is their current utility (because of second part of F). This holds because if nobody in the first round with rebel exists then the strategy of the pool leader in the second round is censor. We say "at most"

because maybe some players have chosen rebel in the second round and u_x^i is increasing in x.

- Regarding another player j who has chosen capitulate in the first round regardless what this player has chosen in the second round : if they choose (rebel, M) where M is an arbitrary strategy then their utility in the first round will become zero that is not higher than their current utility $u_{y-s_i}^j$ and their utility in the second round will remain the same.

  If they choose (capitulate,M) where M is an arbitrary strategy then their utility will remain the same.

□

Note that:

- the equilibria described in Theorem 18 hold even without the assumption that if the pool leader has chosen notcensor it will choose notcensor again and if a pool member has chosen rebel it will choose rebel again.

- we need that there exists unique i such that $s_{i,1} = \text{rebel}$. The fact that there exists i such that $s_{i,1} = \text{rebel}$ is not enough because of the following reason. According to X and Y the pool members and the pool leader choose rebel and notcensor respectively even if just one member has chosen rebel in the previous round. No more than one members are needed to sacrifice their income for signaling.

  Thus in an equilibrium we cannot have more than one member who chooses rebel in the first round. Otherwise a member with rebel could increase its utility by changing its strategy to capitulate, because in this way it would not lose its income for this round and the revolution would still happen in the next round. If X and Y demanded at least x members to choose rebel then we would have equilibria where x are the members choose rebel in the first round.

The following theorem states that if there is a signal and enough players who follow the signal the revolution will happen. Intuitively we prove that if there is a player who chooses rebel in the first round and enough players who follow the signal and choose X in the second round then there is no equilibrium where the pool leader prevents the revolution and chooses censor in the second round.

Theorem 19. *(I) Assuming that u_x^P is increasing in x, if $u'_P > u^P_{y-\alpha}$, where $a = \sum_{i:s_{i,2}=X} s_i$, $\exists i$ such that $s_{i,1} = \mathsf{rebel}$ then there is no equilibrium where $s_{P,1} = \mathit{censor}$ and $s_{P,2} = \mathsf{censor}$ or the pool leader chooses a strategy that ends up to* censor *in the second round. (II) Assuming that u_x^i is increasing in x, if there is a player for whom it holds $u'_i > 2u_y^i$ and there is a player j for whom it holds $s_{j,2} = X$ and $s_{P,2} = \mathsf{notcensor}$ or the pool leader chooses a strategy that ends up to* notcensor *then there is no equilibrium where in the second round the utility of a member i is not u'_i.*

Proof. (I) Let us assume that there exists such an equilibrium. Then the utility of the pool leader will be smaller than $u_y^P + u^P_{y-\alpha}$. If they change their strategy to notcensor in the second round, their utility for the first round will not change and their utility for the second round will become u'^P which is strictly higher.

(II) Let us assume that there exists such an equilibrium. Then in the first round there is no player with rebel and in the second round there is no player with rebel or with strategy that ends up to rebel. The utility of player i in this case is at most $2u_y^i$. If this player changes their strategy to rebel in the first round their utility will become u'_i which is strictly higher. □

6.3.3.2 Multiple Rounds-Single Pool Game with Adaptive Strategies

We can extend theorem 18 so that it captures the above game extended in multiple rounds. If the game lasts k rounds then we can adapt theorem 18 so that (i) the revolution happens in round $j+1$ (ii) for the player who chooses rebel it holds $D_{j,k} = u'_i \geq \frac{k}{k-j} u_y^i$ (instead of $u'_i \geq 2u_y^i$) and (iii) X, Y strategies are adapted as follows:

$X_j = \mathsf{rebel}$ if $\exists i$ such that $s_{i,1} = s_{i,2} = \ldots = s_{i,j} = \mathsf{rebel} \cap D_{j,k}$ and capitulate otherwise.

$Y_j = \mathsf{notcensor}$ if $\exists i$ such that $s_{i,1} = s_{i,2} = \ldots = s_{i,j} = \mathsf{rebel} \cap D_{j,k}$ and censor otherwise.

Note that $D_{j,k}$ represents the fact that it is more profitable for a player to rebel for some rounds and get zero rewards compared to capitulating if at the end they manage to transform the system to a new state with a new pool.

6.3.3.3 Two Rounds-Two Pools Game with Adaptive Strategies

We extend the previous two round game so that we examine in each round a different pool. In more detail, there are two pools with n players each one active in the first and second round respectively. Recall that this captures the scenario where a player of the first pool (which is selected to get dissolved) gives a signal by choosing rebel and in the second round the members of the second pool (which is selected to get dissolved) follow the signal, choose rebel and the revolution takes place. Let P_1, P_2 be the pool leaders of the pools and y_1, y_2 the sums of the stake that belong to all the members and the pool leader of the first and the second pool respectively.

Utilities: the utilities of the players are defined as follows:

- the utility of a player i of the first pool during the first round is defined in the same way as the previous game and in the second round is either $u^i_{y_1}$ or u'_i. In the second round the utility of player i becomes u'_i if:
  1. the pool leader of the second pool chooses notcensor and there is a player of the second pool that chooses rebel.
  2. the utility of player i in the first round is u'_i.
- the utility of the players of the second pool during the first round is $u^i_{y_2}$. The utility of the players of the second pool during the second round is defined as before except when during the first round P_1 chose notcensor and there was a player of the first pool with rebel. In that case the utility will be u'_i.

The utilities of the pool leaders are defined in the same way as the utilities of the members described above.

Let $X = \text{rebel}$ if $\exists i$ such that $s_{i,1} = \text{rebel} \cap D$ and capitulate otherwise.

Let $Y = \text{notcensor}$ if $\exists i$ such that $s_{i,1} = \text{rebel} \cap D$ and censor otherwise. Let D be the event that $u'_i \geq 2u^i_{y_1}$.

Theorem 20. *Assuming that for all i u^P_x and u^i_x is increasing in x and for all i and x it holds $u^i_x \geq 0$ then the following are Nash equilibria: $s_{P_1} = \text{censor}$, $s_{P_2} = Y$ and $K \cap L \cap J \cap F$ where K is the event that $u'_{P_2} \geq u^{P_2}_{y_2 - \alpha}$ where $a = \sum_{i:s_{i,2}=X} s_i$, J the event that $\exists i, j$ such that $s_{i,2} = s_{j,2} = X$, L the event that there exists unique i such that $s_{i,1} =$ rebel and F the event that $(s_{i,1} = rebel) \Rightarrow ((u'^{P_1} \leq u^{P_1}_{y_1 - s_i}) \cap D)$.*

Proof. Let i be the player who chose rebel in the first round.

1. If P_1 changes their strategy to notcensor then their utility will change from $u^{P_1}_{y_1-s_i} + u'_{P_1}$ to $2u'^{P_1}$ which is not higher because of the first part of F.

2. If i changes their strategy to capitulate then their utility will change from u'_i to at most $2u^i_{y_1}$ which is not higher because of F.

3. If a player j from the first pool changes his strategy from capitulate to rebel then his utility will change from $u^{P_1}_{y_1-s_i} + u'_j$ to which is not higher.

4. If P_2 changes their strategy from Y to censor or to a strategy that ends up to censor then their utility will change from $u^{P_2}_{y_2} + u'^{P_2}$ to at most $u^{P_2}_{y_2} + u^{P_2}_{y_2-\alpha}$ which is not higher. If this pool leader changes their strategy from Y to something that ends up to notcensor then their utility will remain the same.

5. If a player of the second pool who has chosen X changes their strategy their utility will remain the same because there is at least one more player with X.

6. If a player of the second pool with a different strategy from X changes their strategy then their utility will remain the same because there exists a player with X and pool leader has chosen Y.

□

Chapter 7

Conclusions

In this thesis we study blockchain protocols from a game theoretic perspective. We study how rational participants who want to maximize their utility react in terms of following the instructions of the protocol and retaining the decentralised nature of the protocol. The first one is important because it guarantees that the protocol will be executed as intended in case the participants are rational, and the second one because decentralization is one of the most important features of blockchain protocols. In general, the research area that combines game theory and blockchain protocols is very active, as blockchain protocols are used in cryptocurrencies and thus inherently provide rewards that determine the choices of rational users. In the previous chapters we have referred to many works that exist in this area.

Summarizing our contribution:

- we propose a framework for examining the incentives in blockchain protocols, and we use it to examine the incentives in the Bitcoin [100] and the Fruitchain blockchain protocol [108]. Moreover we provide a sufficient condition for a blockchain protocol to satisfy our notion of equilibrium. These results are based on our paper [81].

- we propose a reward mechanism that disincentivizes the formation of huge pools in proof of stake blockchain protocols ensuring a level of decentralization. These results are based on our paper [29] published in the IEEE European Symposium on Security and Privacy 2020 and the extended version of this paper in [28]. The reward mechanism that was implemented in the incentivised testnet and the Shelley update on the Cardano mainnet

launched by the company IOHK [3] was based on these results.

- we propose an anti-censorship mechanism that disincnetivizes the pools which run a blockchain protocol to form a cartel and in this way prevent other pools to be created and enter the market.

Adding to the research directions that we give in the previous chapters, some research problems that arise from this thesis and we leave as future work are the following:

- as we discussed in Chapter 5, our reward function does not take into account how well a pool performs. Note that when our reward sharing mechanism is deployed in a proof of stake protocol, it is important to take into account also this aspect, because the pool leaders should have incentives to produce blocks with new transactions when they get elected. To address this, we propose two solutions in Subsection 5.13.1. We leave as future work the formal treatment of the solutions we describe.

- our reward mechanism is designed to be deployed in proof of stake blockchain protocols. An interesting research direction is to examine how we can adapt it to ensure a level of decentralization in a proof of work blockchain protocol, when verifying transactions is a non negligible cost for miners. As we have mentioned in Chapter 5, we leave as future work to prove that in the case when verifying transactions is a non negligible cost for miners, the joint startegy of forming just one huge pool is EVP in the Bitcoin [100] and the Fruitchain blockchain protocol [108].

- one more research direction is to examine what guarantees our reward mechanism in Chapter 5 can provide, when our setting includes both myopic and non-myopic rational players.

- one other potential direction is to examine if the Nash equilibria we describe in Chapter 5 are unique.

- one other research direction that arises from Chapter 4 is to examine if the Fruitchain blockchain protocol [108] remains EVP for coalitions including up to $n-1$ players and utility based on absolute rewards minus absolute cost, when the difficulty in mining is not fixed. Recall that Bitcoin is not EVP in this case.

Bibliography

[1] Bitcoin wiki. `https://en.bitcoin.it/wiki/Main_Page`.

[2] Cloud mining. `https://en.bitcoin.it/wiki/Cloud_mining`.

[3] Iohk. `https://iohk.io/`.

[4] Pool distribution. `https://btc.com/stats/pool`.

[5] *Algorithmic Game Theory*. Cambridge University Press, 2007. Noam Nisan, Tim Roughgarden, Eva Tardos, Vijay V. Vazirani, (Eds.). doi:10.1017/CBO9780511800481.

[6] Ittai Abraham, Lorenzo Alvisi, and Joseph Y. Halpern. Distributed computing meets game theory: Combining insights from two fields. *SIGACT News*, 42(2):69–76, June 2011.

[7] Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *Proceedings of the Twenty-fifth Annual ACM Symposium on Principles of Distributed Computing*, PODC '06, pages 53–62, New York, NY, USA, 2006. ACM.

[8] Ittai Abraham, Danny Dolev, and Joseph Y. Halpern. An almost-surely terminating polynomial protocol for asynchronous byzantine agreement with optimal resilience. In *Proceedings of the Twenty-seventh ACM Symposium on Principles of Distributed Computing*, PODC '08, pages 405–414, New York, NY, USA, 2008. ACM.

[9] Ittai Abraham, Danny Dolev, and Joseph Y. Halpern. Distributed protocols for leader election: A game-theoretic perspective. In Yehuda Afek, editor, *Distributed Computing*, pages 61–75, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[10] Amitanand S. Aiyer, Lorenzo Alvisi, Allen Clement, Mike Dahlin, Jean-Philippe Martin, and Carl Porth. Bar fault tolerance for cooperative services. In *Proceedings of the Twentieth ACM Symposium on Operating Systems Principles*, SOSP '05, pages 45–58, New York, NY, USA, 2005. ACM.

[11] Nick Arnosti and S. Matthew Weinberg. Bitcoin: A natural oligopoly. *CoRR*, abs/1811.08572, 2018.

[12] Robert J. Aumann. *Acceptable Points in General Cooperative n-Person Games. Contributions to the Theory of Games (AM-40)*, volume 4, pages 287–324. Albert William Tucker, Robert Duncan Luce, Princeton: Princeton University Press, 1959. Book DOI: `https://doi.org/10.1515/9781400882168`.

[13] David Austen-Smith and Jeffrey S Banks. Cheap talk and burned money. *Journal of Economic Theory*, 91(1):1–16, 2000.

[14] Adam Back. Hashcash. `http://www.cypherspace.org/hashcash`, 1997.

[15] Kenneth Baclawski. *Introduction to Probability with R.* Chapman & Hall/CRC Texts in Statistical Science. CRC Press, 2008.

[16] Christian Badertscher, Juan Garay, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. But why does it work? a rational protocol design treatment of bitcoin. In Jesper Buus Nielsen and Vincent Rijmen, editors, *Advances in Cryptology – EUROCRYPT 2018*, pages 34–65, Cham, 2018. Springer International Publishing.

[17] Suguman Bansal. *Reasoning about incentive compatibility.* POPL 2016 Student Research Competition, 2016.

[18] Mihir Bellare and Phillip Rogaway. Random oracles are practical: A paradigm for designing efficient protocols. CCS '93, pages 62–73, New York, NY, USA, 1993. Association for Computing Machinery.

[19] Iddo Bentov, Ariel Gabizon, and Alex Mizrahi. Cryptocurrencies without proof of work. *CoRR*, abs/1406.5694, 2014.

[20] Iddo Bentov, Pavel Hub'avcek, Tal Moran, and Asaf Nadler. Tortoise and hares consensus: the meshcash framework for incentive-compatible, scalable cryptocurrencies. *IACR Cryptology ePrint Archive*, 2017:300, 2017.

[21] Iddo Bentov, Rafael Pass, and Elaine Shi. Snow white: Provably secure proofs of stake. *IACR Cryptology ePrint Archive*, 2016:919, 2016.

[22] B.Douglas Bernheim, Bezalel Peleg, and Michael D Whinston. Coalition-proof nash equilibria i. concepts. *Journal of Economic Theory*, 42(1):1 – 12, 1987.

[23] Umang Bhaskar, Lisa Fleischer, Darrel Hoy, and Cien-Chung Huang. Equilibria of Atomic Flow Games are not Unique. *SODA*, pages 748–757, 2009.

[24] Umang Bhaskar and Phani Raj Lolakapuri. Equilibrium computation in atomic splittable routing games with convex cost functions. *arXiv preprint arXiv:1804.10044*, 2018.

[25] Bruno Biais, Christophe Bisiere, Matthieu Bouvard, and Catherine Casamatta. *The Blockchain Folk Theorem.* Swiss Finance Institute Research Paper No. 17-75, 2018.

[26] Joseph Bonneau. Why buy when you can rent? - bribery attacks on bitcoin-style consensus. In Jeremy Clark, Sarah Meiklejohn, Peter Y. A. Ryan, Dan S. Wallach, Michael Brenner, and Kurt Rohloff, editors, *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers*, volume 9604 of *Lecture Notes in Computer Science*, pages 19–26. Springer, 2016.

[27] Joseph Bonneau, Jeremy Clark, and Steven Goldfeder. On bitcoin as a public randomness source. *IACR Cryptol. ePrint Arch.*, 2015:1015, 2015.

[28] Lars Brünjes, Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. Reward sharing schemes for stake pools. *CoRR*, abs/1807.11218, 2018.

[29] Lars Brünjes, Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. Reward sharing schemes for stake pools. In *2020 IEEE European Symposium on Security and Privacy*, pages 256–275, Los Alamitos, CA, USA, sep 2020. IEEE Computer Society.

[30] Vitalik Buterin and Virgil Griffith. Casper the friendly finality gadget. *CoRR*, abs/1710.09437, 2017.

[31] Ran Canetti. Security and composition of multiparty cryptographic protocols. *Journal of Cryptology*, 13(1):143–202, Jan 2000.

[32] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. Cryptology ePrint Archive, Report 2000/067, 2000. `https://eprint.iacr.org/2000/067`.

[33] Ran Canetti. Universally composable security: A new paradigm for cryptographic protocols. In *Proceedings of the 42Nd IEEE Symposium on Foundations of Computer Science*, FOCS '01, pages 136–145, Washington, DC, USA, 2001. IEEE Computer Society.

[34] Miles Carlsten, Harry A. Kalodner, S. Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In Edgar R. Weippl, Stefan Katzenbeisser, Christopher Kruegel, Andrew C. Myers, and Shai Halevi, editors, *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, pages 154–167. ACM, 2016.

[35] George Casella and Roger L Berger. *Statistical Inference*. Duxbury advanced series in statistics and decision sciences. Thomson Learning, 2002.

[36] Xi Chen, Christos H. Papadimitriou, and Tim Roughgarden. An axiomatic approach to block rewards. In *Proceedings of the 1st ACM Conference on Advances in Financial Technologies, AFT 2019, Zurich, Switzerland, October 21-23, 2019*, pages 124–131. ACM, 2019.

[37] David R. Clark. A note on the upper-truncated pareto distribution. *2013 Enterprise Risk Management Symposium*.

[38] Decred Contributors. Decred research overview. https://docs.decred.org/research/overview/, 2018.

[39] Russell Cooper, Douglas V DeJong, Robert Forsythe, and Thomas W Ross. Communication in coordination games. *The Quarterly Journal of Economics*, 107(2):739–771, 1992.

[40] Vincent P Crawford and Joel Sobel. Strategic information transmission. *Econometrica: Journal of the Econometric Society*, pages 1431–1451, 1982.

[41] Varsha Dani, Mahnush Movahedi, Yamel Rodriguez, and Jared Saia. Scalable rational secret sharing. In Cyril Gavoille and Pierre Fraigniaud, editors, *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing, PODC 2011, San Jose, CA, USA, June 6-8, 2011*, pages 187–196. ACM, 2011.

[42] Bernardo David, Peter Gaži, Aggelos Kiayias, and Alexander Russell. Ouroboros praos: An adaptively-secure, semi-synchronous proof-of-stake protocol. Cryptology ePrint Archive, Report 2017/573, 2017. `http://eprint.iacr.org/2017/573`. To appear at EUROCRYPT 2018.

[43] John R. Douceur. The sybil attack. In *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, IPTPS '01, pages 251–260, London, UK, UK, 2002. Springer-Verlag.

[44] Devdatt Dubhashi and Alessandro Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, USA, 1st edition, 2009.

[45] Cynthia Dwork and Moni Naor. Pricing via processing or combatting junk mail. In Ernest F. Brickell, editor, *Advances in Cryptology — CRYPTO' 92*, pages 139–147, Berlin, Heidelberg, 1993. Springer Berlin Heidelberg.

[46] Kfir Eliaz. Fault tolerant implementation. *The Review of Economic Studies*, 69, 2002.

[47] Ittay Eyal. The miner's dilemma. *CoRR*, abs/1411.7099, 2014.

[48] Ittay Eyal and Emin Gun Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Angelos D. Keromytis, editor, *Financial Cryptography*, volume 7397 of *Lecture Notes in Computer Science*. Springer, 2014.

[49] Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In Nicolas Christin and Reihaneh Safavi-Naini, editors, *Financial Cryptography and Data Security - 18th International Conference, FC 2014, Christ Church, Barbados, March 3-7, 2014, Revised Selected Papers*, volume 8437 of *Lecture Notes in Computer Science*, pages 436–454. Springer, 2014.

[50] Giulia Fanti, Leonid Kogan, Sewoong Oh, Kathleen Ruan, Pramod Viswanath, and Gerui Wang. Compounding of wealth in proof-of-stake cryptocurrencies, 2018.

[51] Joseph Farrell. Cheap talk, coordination, and entry. *The RAND Journal of Economics*, pages 34–39, 1987.

[52] Joseph Farrell and Matthew Rabin. Cheap talk. *Journal of Economic perspectives*, 10(3):103–118, 1996.

[53] Chaim Fershtman and Kenneth L Judd. Equilibrium incentives in oligopoly. *American Economic Review*, 77(5):pp. 927 – 940, 1987.

[54] Amos Fiat, Anna Karlin, Elias Koutsoupias, and Christos Papadimitriou. Energy equilibria in proof-of-work mining. In *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, pages 489–502, New York, NY, USA, 2019. ACM.

[55] Amos Fiat, Elias Koutsoupias, Katrina Ligett, Yishay Mansour, and Svetlana Olonetsky. Beyond myopic best response (in cournot competition). *Games and Economic Behavior*, 2013.

[56] Ben Fisch, Rafael Pass, and Abhi Shelat. Socially optimal mining pools. In Nikhil R. Devanur and Pinyan Lu, editors, *Web and Internet Economics - 13th International Conference, WINE 2017, Bangalore, India, December 17-20, 2017, Proceedings*, volume 10660 of *Lecture Notes in Computer Science*, pages 205–218. Springer, 2017.

[57] Georg Fuchsbauer, Jonathan Katz, and David Naccache. Efficient rational secret sharing in standard communication networks. In Daniele Micciancio, editor, *Theory of Cryptography, 7th Theory of Cryptography Conference, TCC 2010, Zurich, Switzerland, February 9-11, 2010. Proceedings*, volume 5978 of *Lecture Notes in Computer Science*, pages 419–436. Springer, 2010.

[58] Juan Garay, Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, FOCS '13, pages 648–657, Washington, DC, USA, 2013. IEEE Computer Society.

[59] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In Elisabeth Oswald and Marc Fischlin, editors, *Advances in Cryptology - EUROCRYPT 2015*, pages 281–310, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[60] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol with chains of variable difficulty. In Jonathan Katz and Hovav Shacham, editors, *Advances in Cryptology – CRYPTO 2017*, pages 291–323, Cham, 2017. Springer International Publishing.

[61] Dino Gerardi. Unmediated communication in games with complete and incomplete information. *Journal of Economic Theory*, 114(1):104 – 131, 2004.

[62] Arthur Gervais, Ghassan O. Karame, Vedran Capkun, and Srdjan Capkun. Is bitcoin a decentralized currency? *IEEE Security Privacy*, 12(3):54–60, May 2014.

[63] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. On the security and performance of proof of work blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, pages 3–16, New York, NY, USA, 2016. ACM.

[64] Ian Grigg. Eos, an introduction. `https://eos.io/documents/EOS\_An\_Introduction.pdf`, 2017.

[65] Adam Groce and Jonathan Katz. Fair computation with rational players. In David Pointcheval and Thomas Johansson, editors, *Advances in Cryptology – EUROCRYPT 2012*, pages 81–98, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[66] Adam Groce, Jonathan Katz, Aishwarya Thiruvengadam, and Vassilis Zikas. Byzantine agreement with a rational adversary. In Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, editors, *Automata, Languages, and Programming*, pages 561–572, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[67] Cyril Grunspan and Ricardo Pérez-Marco. On profitability of selfish mining. *CoRR*, abs/1805.08281, 2018.

[68] Önder Gürcan, Antonella Del Pozzo, and Sara Tucci-Piergiovanni. On the bitcoin limitations to deliver fairness to users. In Hervé Panetto, Christophe Debruyne, Walid Gaaloul, Mike Papazoglou, Adrian Paschke, Claudio Agostino Ardagna, and Robert Meersman, editors, *On the Move to Meaningful Internet Systems. OTM 2017 Conferences*, pages 589–606, Cham, 2017. Springer International Publishing.

[69] Joseph Y. Halpern and Xavier Vilaça. Rational consensus: Extended abstract. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, pages 137–146, New York, NY, USA, 2016. ACM.

[70] Jr. Harvey S. James. *Incentive compatibility*. Encyclopedia Britannica, inc., 4 2014. Encyclopedia Britannica `https://www.britannica.com/topic/incentive-compatibility`.

[71] Charlie Hou, Mingxun Zhou, Yansquir Ji, Phil Daian, Florian Tramer, Giulia Fanti, and Ari Juels. Squirrl: Automating attack discovery on blockchain incentive mechanisms with deep reinforcement learning, 2019.

[72] Edward W Felten Joshua A Kroll, Ian C Davey. The economics of bitcoin mining, or bitcoin in the presence of adversaries. In *Proceedings of WEIS*, 2013.

[73] Ari Juels and John G. Brainard. Client puzzles: A cryptographic countermeasure against connection depletion attacks. In *NDSS*. The Internet Society, 1999.

[74] Dimitris Karakostas, Aggelos Kiayias, Christos Nasikas, and Dionysis Zindros. Cryptocurrency egalitarianism: A quantitative approach. Tokenomics, International Conference on Blockchain Economics, Security and Protocols, 2019.

[75] Jonathan Katz. Bridging game theory and cryptography: Recent results and future directions. In Ran Canetti, editor, *Theory of Cryptography*, pages 251–272, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[76] Jonathan Katz and Yehuda Lindell. *Introduction to Modern Cryptography (Chapman and Hall/Crc Cryptography and Network Security Series)*. Chapman and Hall/CRC, 2007.

[77] Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. Universally composable synchronous computation. In Amit Sahai, editor, *Theory of Cryptography*, pages 477–498, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[78] Yoav Shoham Kevin Leyton-Brown. *Essentials of Game Theory: A Concise Multidisciplinary Introduction (Synthesis Lectures on Artificial Intelligence and Machine Learning)*. Morgan & Claypool Publishers, 2008.

[79] Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis. Blockchain mining games. In Vincent Conitzer, Dirk Bergemann, and Yiling Chen, editors, *Proceedings of the 2016 ACM Conference on Economics and Computation, EC '16, Maastricht, The Netherlands, July 24-28, 2016*, pages 365–382. ACM, 2016.

[80] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. Cryptology ePrint Archive, Report 2016/889, 2016. `http://eprint.iacr.org/2016/889`.

[81] Aggelos Kiayias and Aikaterini-Panagiota Stouka. Coalition-safe equilibria with virtual payoffs. *CoRR*, abs/2001.00047, 2020.

[82] Abhiram Kothapalli, Andrew Miller, and Nikita Borisov. Smartcast: An incentive compatible consensus protocol using smart contracts. In Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y.A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, and Markus Jakobsson, editors, *Financial Cryptography and Data Security*, pages 536–552, Cham, 2017. Springer International Publishing.

[83] Yujin Kwon, Jian Liu, Minjeong Kim, Dawn Song, and Yongdae Kim. Impossibility of full decentralization in permissionless blockchains. *CoRR*, abs/1905.05158, 2019.

[84] Leslie Lamport, Robert Shostak, and Marshall Pease. The byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[85] Dan Larimer. Delegated proof-of-stake consensus. `https://bitshares.org/technology/delegated-proof-of-stake-consensus/`, accessed 21.3.2018, 2018.

[86] Nikos Leonardos, Stefanos Leonardos, and Georgios Piliouras. Oceanic games: Centralization risks and incentives in blockchain mining. *CoRR*, abs/1904.02368, 2019.

[87] Stefanos Leonardos, Daniel Reijsbergen, and Georgios Piliouras. Presto: A systematic framework for blockchain consensus protocols, 2019.

[88] Matt Lepinski, Silvio Micali, Chris Peikert, and Abhi Shelat. Completely fair SFE and coalition-safe cheap talk. In Soma Chaudhuri and Shay Kutten, editors, *Proceedings of the Twenty-Third Annual ACM Symposium on Principles of Distributed Computing, PODC 2004, St. John's, Newfoundland, Canada, July 25-28, 2004*, pages 1–10. ACM, 2004.

[89] Matt Lepinski, Silvio Micali, Chris Peikert, and Abhi Shelat. Completely fair sfe and coalition-safe cheap talk. In *Proceedings of the Twenty-third Annual ACM Symposium on Principles of Distributed Computing*, PODC '04, pages 1–10, New York, NY, USA, 2004. ACM.

[90] Yoad Lewenberg, Yonatan Sompolinsky, and Aviv Zohar. Inclusive block chain protocols. In Rainer Böhme and Tatsuaki Okamoto, editors, *Financial Cryptography and Data Security*, pages 528–547, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

[91] Kevin Liao and Jonathan Katz. Incentivizing blockchain forks via whale transactions. In Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y.A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, and Markus Jakobsson, editors, *Financial Cryptography and Data Security*, pages 264–279, Cham, 2017. Springer International Publishing.

[92] Jian Liu, Wenting Li, Ghassan O. Karame, and N. Asokan. Toward fairness of cryptocurrency payments. *IEEE Security Privacy*, 16(3):81–89, May 2018.

[93] Ziyao Liu, Nguyen Cong Luong, Wenbo Wang, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. A survey on applications of game theory in blockchain, 2019.

[94] Loi Luu, Yaron Velner, Jason Teutsch, and Prateek Saxena. Smartpool: Practical decentralized pooled mining. In *26th USENIX Security Symposium (USENIX Security 17)*, pages 1409–1426, Vancouver, BC, August 2017. USENIX Association.

[95] Anna Lysyanskaya and Nikos Triandopoulos. Rationality and adversarial behavior in multi-party computation. In Cynthia Dwork, editor, *Advances in Cryptology - CRYPTO 2006*, pages 180–197, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[96] Andreu Mas-Colell, Jerry R Green, and Michael D. Whinston. *Microeconomic Theory*. Oxford University Press, 1995.

[97] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. A fistful of bitcoins: Characterizing payments among men with no names. In *Proceedings of the 2013 Conference on Internet Measurement Conference*, IMC '13, pages 127–140, New York, NY, USA, 2013. ACM.

[98] Silvio Micali. ALGORAND: The efficient and democratic ledger, 2016.

[99] Roger B. Myerson. *Game Theory: Analysis of Conflict*. Cambridge, Mass: Harvard University Press, 1991.

[100] Satoshi Nakamoto. *Bitcoin: A Peer-to-Peer Electronic Cash System*, 2008. `http://bitcoin.org/bitcoin.pdf`.

[101] Arvind Narayanan, Joseph Bonneau, Edward Felten, Andrew Miller, and Steven Goldfeder. *Bitcoin and Cryptocurrency Technologies: A Comprehensive Introduction*. Princeton University Press, USA, 2016.

[102] Kartik Nayak, Srijan Kumar, Andrew Miller, and Elaine Shi. Stubborn mining: Generalizing selfish mining and combining with an eclipse attack. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*, pages 305–320, March 2016.

[103] Noam Nisan, Tim Roughgarden, Eva Tardos, and Vijay V Vazirani. *Algorithmic game theory*. Cambridge University Press, 2007.

[104] Ariel Orda, Raphael Rom, and Nahum Shimkin. Competitive Routing in Multiuse Communication Networks. *IEEE/ACM Transactions on Networking*, 1(5):510–521, 1993.

[105] Guillermo Owen. Game theory academic press. *San Diego*, 1995.

[106] Rafael Pass and Joe Halpern. Game theory with costly computation: Formulation and application to protocol security. In *Proceedings of the Behavioral and Quantitative Game Theory: Conference on Future Directions*, BQGT '10, pages 89:1–89:1, New York, NY, USA, 2010. ACM.

[107] Rafael Pass, Lior Seeman, and Abhi Shelat. Analysis of the blockchain protocol in asynchronous networks. In Jean-Sébastien Coron and Jesper Buus Nielsen, editors, *Advances in Cryptology – EUROCRYPT 2017*, pages 643–673, Cham, 2017. Springer International Publishing.

[108] Rafael Pass and Elaine Shi. Fruitchains: A fair blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing*, PODC '17, pages 315–324, New York, NY, USA, 2017. Association for Computing Machinery.

[109] Marshall Pease, Robert Shostak, and Leslie Lamport. Reaching agreement in the presence of faults. *J. ACM*, 27(2):228–234, April 1980.

[110] Oran Richman and Nahum Shimkin. Topological Uniqueness of the Nash Equilibrium for Selfish Routing with Atomic Users. *Mathematics of Operations Research*, 32(1):215–232, 2007.

[111] Ronald L. Rivest, Adi Shamir, and David A. Wagner. Time-lock puzzles and timed-release crypto. Technical report, Cambridge, MA, USA, 1996.

[112] Matteo Romiti, Aljosha Judmayer, Alexei Zamyatin, and Bernhard Haslhofer. A deep dive into bitcoin mining pools: An empirical analysis of mining shares, 2019.

[113] Dorit Ron and Adi Shamir. Quantitative analysis of the full bitcoin transaction graph. In Ahmad-Reza Sadeghi, editor, *Financial Cryptography and Data Security*, pages 6–24, Berlin, Heidelberg, 2013. Springer Berlin Heidelberg.

[114] Meni Rosenfeld. Analysis of bitcoin pooled mining reward systems. *CoRR*, abs/1112.4980, 2011.

[115] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. Optimal selfish mining strategies in bitcoin. In Jens Grossklags and Bart Preneel, editors, *Financial Cryptography and Data Security*, pages 515–532, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.

[116] Thomas C Schelling. An essay on bargaining. *The American Economic Review*, 46(3):281–306, 1956.

[117] Thomas C Schelling. *The strategy of conflict*. Harvard university press, 1980.

[118] Okke Schrijvers, Joseph Bonneau, Dan Boneh, and Tim Roughgarden. Incentive compatibility of bitcoin mining pool reward functions. In Jens Grossklags and Bart Preneel, editors, *Financial Cryptography and Data Security*, pages 477–498, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.

[119] Steven D. Sklivas. The strategic choice of managerial incentives. *RAND Journal of Economics*, 18(3):pp. 452 – 458, 1987.

[120] Itay Tsabary and Ittay Eyal. The gap game. *CoRR*, abs/1805.05288, 2018.

[121] Amparo Urbano and José Enrique Vila. Unmediated communication in repeated games with imperfect monitoring. *Games and Economic Behavior*, 46(1):143–173, 2004.

[122] John B Van Huyck, Raymond C Battalio, and Richard O Beil. Tacit coordination games, strategic uncertainty, and coordination failure. *The American Economic Review*, 80(1):234–248, 1990.

[123] John Vickers. Delegation and the theory of the firm. *Economic Journal*, 95 (Suppl.):pp. 138 – 147, 1985.

[124] William Vickrey. Counterspeculation, auctions, and competitive sealed tenders. *The Journal of finance*, 16(1):8–37, 1961.

[125] John Ross Wallrabenstein and Chris Clifton. Equilibrium concepts for rational multiparty computation. In Sajal K. Das, Cristina Nita-Rotaru, and Murat Kantarcioglu, editors, *Decision and Game Theory for Security*, pages 226–245, Cham, 2013. Springer International Publishing.