Coalition-Safe Equilibria with Virtual Payoffs

Coalition-Safe Equilibria with Virtual Payoffs

Aggelos Kiayias
University of Edinburgh and IOHK
Edinburgh, United Kingdom
akiayias@inf.ed.ac.uk

Aikaterini-Panagiota Stouka
University of Edinburgh
Edinburgh, United Kingdom
A.Stouka@ed.ac.uk

ABSTRACT

Consider a set of participants invited to execute a protocol Π. The protocol will incur some cost to run while in the end (or at regular intervals), it will populate and update local bookkeeping tables that assign *virtual* rewards to participants. Each participant aspires to offset the costs of participation by these virtual payoffs that are provided in the course of the protocol and are assumed to be accepted as forms of payment. In this setting, we introduce and study a notion of coalition-safe equilibria. In particular, we consider a strategic coalition of participants that is centrally coordinated and potentially deviates from Π with the objective to increase its utility with respect to the view of *at least one* of the other participants. The protocol Π is called a coalition-safe equilibrium with virtual payoffs (EVP) if no such protocol deviation exists. We apply our notion to study incentives in blockchain protocols.

Compared to prior work, our framework has the advantages that it simultaneously (i) takes into account that each participant may have a divergent view of the rewards given to the other participants, as the reward mechanism employed is subject to consensus among participants (and our notion is well defined independently of whether the underlying protocol achieves consensus or not) (ii) accounts for the stochastic nature of these protocols by enforcing the equilibrium condition to hold with overwhelming probability.

We use our framework to provide a unified picture of incentives in the Bitcoin blockchain, for absolute and relative rewards based utility functions. Importantly, we prove that organizing all miners into a single dictatorial pool is an EVP in the setting of non-zero transaction verification costs for coalitions of up to $n-1$ participants. In addition we prove novel results regarding incentives of the Fruitchain blockchain protocol [PODC 2017] showing that the equilibrium condition holds for coalitions up to $n-1$ participants for absolute rewards based utility functions and less than $n/2$ for relative rewards based utility functions, with the latter result holding for any "weakly fair" blockchain protocol, a new property that we introduce and may be of independent interest.

CCS CONCEPTS

• **Information systems** → **Incentive schemes**; • **Theory of computation** → **Solution concepts in game theory**; *Algorithmic game theory*; Distributed computing models; • **Security and privacy** → Distributed systems security.

AFT '21, September 26–28, 2021, Arlington, VA, USA

ACM ISBN 978-1-4503-9082-8/21/09.
https://doi.org/10.1145/3479722.3480795

KEYWORDS

incentives, protocols, game theoretic analysis, blockchain

ACM Reference Format:
Aggelos Kiayias and Aikaterini-Panagiota Stouka. 2021. Coalition-Safe Equilibria with Virtual Payoffs. In *3rd ACM Conference on Advances in Financial Technologies (AFT '21), September 26–28, 2021, Arlington, VA, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3479722.3480795

1 INTRODUCTION

A distributed protocol thought as a game, involves a number of participants that engage with each other following a certain strategy profile (e.g., following the protocol or deviating from it) and may incur individual costs as well as rewards. The utility of each participant, which rational participants aspire to maximize, is some compound real-valued function that takes into account the costs incurred and rewards resulting by the interaction. A common characteristic in game theoretic analysis is that costs and rewards are bestowed authoritatively via some infrastructure that is typically external to the game execution. Contrary to this, in this work, we study a game-theoretic setting where rewards are *virtual* and are recorded as an outcome of the interaction of the participants individually in each participant's local view. Thus, while costs are incurred authoritatively as before, rewards are "in the eye of the beholder" and in the end of the interaction two participants may have diverging views about the rewards that each game participant has received, while any single participant P cares fundamentally that all the other participants conclude in their local views that P has received some minimum amount of rewards.

Our motivation comes from the setting of distributed ledgers. These protocols were originally studied as an instance of the state machine replication problem [58] but recently were popularised again due to the introduction of the Bitcoin blockchain protocol [50]. Bitcoin introduced by [50] is a cryptocurrency based on a blockchain protocol that maintains a public ledger containing the history of all transactions. The protocol was formally analyzed in the cryptographic setting in [27, 52]. The main idea behind the protocol is that transactions are organized into blocks and blocks form a chain, as each block contains a fingerprint of the previous block. The longest chain is selected to determine the public ledger. A block is produced when a proof of work puzzle [11, 23, 35, 54] is solved by a participant who is called a miner. The *difficulty* of solving such a puzzle is adapted to the total computational power of the miners such that the average number of blocks per unit of time remain steady. Intuitively solving a puzzle in Bitcoin means computing the hash value of the block with different nonce each time and trying to find a hash value that is smaller than a *target*. Miners receive as reward for each block they mine as follows: (i) a flat reward and (ii) an amount of Bitcoin for each transaction they include in the block. The flat reward gets minted when the block

is produced. The transaction fees are determined and paid by the sender of the transaction.

One distinguishing feature of blockchain protocols is the emphasis they put on the incentives of the participating entities. Classically, consensus [41] was considered in various threat models, such as fail-stop failures or Byzantine. However the incentive and game theoretic aspects of the protocol have received less attention.

In blockchain protocols, the rewards that are bestowed to the participants are not assigned in an authoritative manner by some external entity, but rather are recorded as an outcome of bookkeeping that takes place by the interaction of the participants. In such setting, an important question is whether a strategic coalition of participants has an incentive to follow the protocol or to deviate. The most general form is a "monolithic" coalition (abstracted as an "adversary") that considers deviating from the protocol in a coordinated fashion with the aim to increase the joint utility (the sum of the utilities) of its members.

Different aspects of incentives in Bitcoin were studied in [10, 12, 17, 22, 24, 34, 38, 45, 48] and incentive compatibility for blockchain protocols was studied in the context of a few protocols, see e.g., [6, 14, 53]. With respect to studying the participation in the core blockchain protocol, Kroll et al. in [34] show that a certain modeling of the Bitcoin protocol is a Nash equilibrium, while Eyal and Sirer in [24] show that Bitcoin is not incentive compatible because of a type of attack called selfish mining that works for any level of computational power (for *Nash equilibrium* and *incentive compatibility* definition see related work Section 1.2). Then again, Kiayias et al. in [38] show that there are thresholds of computational power where certain games that abstract Bitcoin have honest behavior as a Nash equilibrium. The above seemingly contradictory results stem from differences in the game theoretic modeling of the underlying blockchain protocol and the utility function that is postulated. In addition, the existing notions of equilibria (cf. Section 1.2 below) do not appear to be sufficient to completely capture the rational behavior of participants. First, given the anticipated long term execution length of such games it is important to consider the variance of utility and thus merely looking at expected utility might be insufficient. Second, the reward mechanism employed is subject to consensus among participants and given that the protocol itself aims to achieve such consensus, each participant may have a divergent view of the rewards given to the other participants.

Thus it is important that the model used to examine the protocol takes into account the possibility of such divergence and the game should be well defined independently of whether the resulting interaction achieves consistency or safety, as such properties should be the result of the rational interaction of participants, not a precondition for it!

1.1 Our Results

Execution model and Utility with Virtual Payoffs: Our model generalizes the execution model of [27] and it is based on the "real-world" protocol execution model of [19–21, 36] with the additional feature that certain operations of the protocol are abstracted as oracles and calling such oracles incurs a certain cost to the callers. In this way, the cost of each participant is solely dependent on participants' actions and aggregates the expenditure that is incurred during the execution. For example in the case of a proof of work [11, 23, 35, 54] blockchain protocol such as Bitcoin this may amount to the number of queries posed to the hash function. At any point of an execution, each participant has a local view regarding the virtual rewards of all participants, including themselves. The key observation for defining utility in our setting is that given that the rewards are virtual, it is not particularly advantageous for a participant to be in a state where according to its own bookkeeping it has collected some rewards; instead what is important, is what *other* participants believe about the participant's rewards. In this way we define two types of reward functions $R^{\max}, R^{\min}$ which will correspondingly give rise to two utility functions ($U^{\max}$ and $U^{\min}$). Note that the cost used in the utility functions is independent of the view of the honest parties. The $R^{\max}$ rewards of a coalition represent the maximum amount of rewards a coalition has received quantified over all *other* participants (which do not belong to the coalition), while $R^{\min}$ is similarly the minimum amount of rewards.

Equilibria with Virtual Payoffs (EVP) . Based on these functions (reward, cost and utility functions), we present a formal notion of approximate Nash equilibrium, called coalition-safe Equilibrium with Virtual Payoffs (EVP). Informally, a protocol Π is an EVP if it guarantees that with overwhelming probability, a rational strategic actor (hence called the *adversary*) who controls a coalition of participants, cannot, by possibly deviating, significantly increase their 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 deviating from Π becomes significantly higher in the view of a single other participant then such protocol will *not be* an EVP. In more details, our notion of equilibrium is defined by examining two independent executions of the protocol in question. In the first execution the adversary controlling a coalition follows the protocol while in the second execution it might deviate in some strategic fashion. In both executions the participants who are not controlled by the adversary (we refer to them as honest participants) follow the protocol [1]. 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 is EVP when with overwhelming probability the $U^{\max}$ utility of the adversary when it deviates is not significantly higher compared to its $U^{\min}$ utility when it follows the protocol. This means that in order for the underlying protocol not to be an EVP, there will be an alternative strategy and an environment with respect to which: the execution where the adversary deviates results, with a non-negligible probability, to a significantly higher utility, in the view of one honest participant, compared to its *lowest* utility in the execution where it follows the protocol (lowest is quantified over the view of *all* the honest participants). In our analysis, we revisit three important utility definitions for blockchain protocols: (i) absolute rewards (ii) absolute rewards minus absolute cost and (iii) relative rewards. With the term absolute rewards we refer to the amount of the rewards that a set of participants accrues by the end of the execution. With the term absolute cost we mean the cost that this set of participants pays

[1] This is in line with the notion of *Nash equilibrium* where all but one (or all except a coalition) follow the indicated strategy, and we examine if one (or a coalition) can increase its utility by choosing a strategy different from the indicated.

during the execution 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. We note that the first and the third type of utility have been considered in a number of previous works, specifically, [34, 53] used the first type and [14, 24, 38] used the third type. In addition the second type was used in [12, 40, 59]. For the formal definition of these functions cf. Definition 2.2.

EVP Analysis of Blockchain Protocols: Using our model we prove positive and negative results regarding the incentives in Bitcoin unifying previous seemingly divergent views on how the protocol operates in terms of incentives, cf. Theorems 3.1,3.2,3.3,3.4,3.5. Specifically, we prove that Bitcoin with fixed target is an EVP with utility based on absolute rewards, and absolute rewards minus absolute costs, while it is not with respect to relative rewards, cf. Figure 1.

We then prove that when the cost of processing the transactions is non-zero then deviating from the Bitcoin protocol and converging all parties into a single mining pool is an EVP according to the utility function based on absolute rewards minus absolute cost (cf. Theorem 4.2). The utility of this strategy dominates the Bitcoin protocol and thus the all Bitcoin miners will prefer to centralize to a single operator entity rather than run the protocol individually.[2]

Next, we prove regarding incentives of Fruitchain, [53], the following new result: when the utility is based on absolute rewards minus absolute cost, the Fruitchain protocol is an EVP against a coalition including even up to *all but one* of the participants (cf. Theorem 6.2). Moreover we define a property called "(t, δ)-weak fairness" that is weaker than "fairness" defined in [53] or "ideal chain quality" described in [27] and the "race-free property" in [14] (for more details cf. Section 5) and is sufficient for proving that a protocol is EVP when the utility is based on relative rewards (cf. Theorem 5.2). This allows us to also prove the following result: when the utility is based on relative rewards, the Fruitchain protocol is EVP in the synchronous setting (cf. Section 2) against any coalition including fewer than half of the number of the participants (assuming participants of equal computational power, cf. Theorem 6.1). Further, we note that the approximation factor in the EVP is merely a constant additive factor. Regarding the level of rewards, in [53] the total rewards V of an execution are derived from: (a) the flat rewards of the fruits (for details regarding what a fruit according to [53] is, cf. Section 6) and (b) the transaction fees from the transactions inside the fruits; in both cases these are distributed evenly among the miners and V is a fixed constant in the whole execution. Our result improves the above in this respect, for both absolute and relative rewards based utilities, as we show that the protocol is an EVP even if rewards are a function of the security parameter or the length of the execution.

We note that our model is synchronous and in our results we consider that the adversary is "static" which means that the participants that form the adversarial coalition are fixed every time at the onset together with the hashing costs that are payable during each round. We will refer to this setting as "static adversary with fixed cost." This type of cost model is consistent with cloud mining [1] where participants establish a contract and they pay a fixed rental fee per time unit.

[2] Arguably this is also observed in the real world where just a handful of mining pools (see, e.g., https://www.blockchain.com/pools) control the system. Presumably the existence of externalities (such as the perception of decentralization and the way this affects the value of Bitcoin) is the a possible reason for not converging to a single mining pool which would have been the preferred choice per our result (that assumes no externalities).

	AbsR/AbsR-C	RelR
Bitcoin fixed target	$n-1$ $^{(*)(c)}$	NO$^{(1)}$
Bitcoin variable target	NO$^{(2)}$	NO$^{(3)}$
Fruitchain	$(n-1)$ $^{(\dagger)}$	$< n/2$

Figure 1: Overview of our results as well as previous results that are consistent with the EVP model regarding whether following the Bitcoin and the Fruitchain protocol is an equilibrium. AbsR stands for a utility based on absolute rewards, AbsR-C for a utility based on absolute rewards minus absolute cost, while RelR stands for a utility based on relative rewards. The function in n specifies the larger coalition size for which the equilibrium stands. "NO" means that following the protocol is not an equilibrium. $^{(1),(3)}$ are derived from [24], $^{(2)}$ is derived from [25]. $^{(*)}$ is related to the results presented in [12, 34] ; with $^{(c)}$ we indicate our result that takes into account the cost of processing the transactions and demonstrates that Bitcoin with fixed target is dominated by the single mining pool strategy. A weaker bound of the $^{(\dagger)}$ result in terms of coalition size (specifically $< n/2$) was shown in [53].

1.2 Other Related Work

Recall that a strategy, which indicates how each participant behaves in the game, is an *ϵ-Nash equilibrium* when the following holds (see e.g., [37]): if all but one of the participants follow their strategy indicated by the strategy profile, the remaining participant has no incentives to deviate from its indicated strategy as well, as its utility can only be increased by a small insignificant amount bounded by ϵ. Extended notions of equilibria capture strategic coalitions as well, cf. [9, 15], giving rise to "Strong" Nash Equilibria. Note that if we show that a blockchain protocol is an ϵ-Nash equilibrium, we know that nobody can increase its utility by more than ϵ, via a unilateral deviation, assuming that everybody else follows the protocol.

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 [4]. In cryptocurrency literature some times the incentive compatibility notion is used as equivalent to the Nash equilibrium notion [43]. 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 [32].

A closely related work that focused on Byzantine Agreement and rational behavior is [30]. Some distinctions between our work and [30] 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 transactions 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 (including those that "break" consensus) as long as the adversarial coalition benefits in the view of one participant.

The "BAR" model introduced in [7] combines Byzantine participants, i.e., participants that can deviate from the protocol arbitrarily, in addition to honest and rational participants. This model includes three types of participants: altruistic, Byzantine and rational. 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 [5]. In "ϵ-(k,t)-robust Nash equilibrium"[5] 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. Note that in our case following [53] 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 [27, 53] 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 [26] 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 [12]. In [12] the basic premise is that the miners aim to maximize their expected revenue and the framework describes a meta-game between just 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 [12] is that we let the coalition 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 [12] focuses exclusively on the incentive compatibility of Bitcoin and only when utility is based on absolute rewards minus absolute cost and does not captures issues such as pooling.

Moreover, a framework for identifying attacks against the incentive schemes of the blockchain protocols is proposed in [33]. In [16], proof of work blockchain protocols are modeled as stochastic games while in [47] a survey of game theoretic applications in the blockchain setting is presented.

As we already mentioned, in [53] the Fruitchain protocol is presented, which preserves the security properties of Bitcoin protocol and satisfies a δ-approximate fairness property (assuming honest majority) that is shown to be enough for incentive compatibility when the utility is based on absolute rewards. In addition, in [53] a definition of approximate Nash equilibrium is described, denoted by"ρ-coalition-safe ϵ-Nash equilibrium" that guarantees protocol conformity with overwhelming probability. Our EVP definition compared to [53] 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 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 *common prefix property* defined in [27]. In high level when a blockchain protocol satisfies *common prefix property*, then all the honest participants' local chains share the same prefix with overwhelming probability. As it has been proved in [27], Bitcoin satisfies common prefix property assuming honest majority. Note that when we study a blockchain protocol from a game theoretic perspective, where we allow rational coalitions to include almost all the participants, we cannot assume honest majority and thus that common prefix property is satisfied (iv) it is well defined also for utilities that can take negative values, such as that one based on absolute rewards minus absolute cost.

Another property related to "fairness" is "t-immunity" in [5]. The last property considers utility as an expectation. Note that the notion of fairness has also been used in [44]. A notion of weak fairness has also been used in [46] for a different purpose. Specifically in [46] fairness refers to exchanges between participants; both or neither of the participants take the other's item.

A closely related work in terms of Bitcoin centralization is [18] which proves that if the Bitcoin reward function is used in a pooled proof of stake setting that is deterministic, then there is no equilibrium with more than one pool. We note that in this paper we examine directly the Bitcoin protocol taking into account the cost of mining and processing transactions and also its stochastic nature. Moreover, our result on Bitcoin centralizing in a single mining pool when transaction processing costs are non-zero is also complementary to the result of Arnosti and Weinberg [8] who prove that deviations in the hashing power costs may also lead to centralization. In their setting centralization takes place as a response to differences in electricity costs; instead in our setting centralization occurs due to the non-negligible cost of transaction processing and the rational desire to share that cost. Finally other works related to the Bitcoin centralization are [28, 42, 60].

For more details regarding the related work see the full version of this paper in [39].

2 OUR MODEL

Our definition of coalition-safe equilibria with virtual payoffs is built on a model of protocol execution that extends the model described in [27], and is based on [19–21, 36]. This model constitutes the basis for analyzing incentives in an arbitrary blockchain protocol Π (but is not necessarily restricted to blockchain protocols). The main components of the model are: a system of interactive Turing machines ITMs ($\mathcal{Z}$,$\mathcal{C}$), a strategic coalition of participants

that abstractly are referred to as the "adversary", $\mathcal{A}$ which is also an ITM, and the ITM instances (ITIs) $P_1, P_2, ..., P_n$ that represent the participants of our protocol that run the blockchain protocol Π. C is the control program that controls the interactions between the ITIs. $\mathcal{Z}$ is the "environment" or in other words the initial Turing machine that represents the external world to the protocol. It gives inputs to the participants and the adversary and it receives outputs from them. The adversary is static and controls a set of t' participants $\{P_{i_1}, ..., P_{i_{t'}}\} \subseteq \{P_1, ..., P_n\}$ in the beginning of the execution. We will use the notation $T = \{P_{i_1}, ..., P_{i_{t'}}\}$ and $S = \{P_1, ..., P_n\}$. Let t be the upper bound on the number of the participants the adversary controls. In the definition of equilibrium we will put forth, we consider executions where adversary follows an arbitrary strategy while the remaining participants follow Π.

The execution is synchronous and is progressing in rounds as in [27], which means that at the end of each round all the honest participants receive all the messages sent from all the other honest participants.

However, compared to [27], instead of just a random oracle on which a cryptographic hash function [3] is modelled, we allow for many oracles, where each oracle represents a cryptographic task, such as issuing a digital signature or processing transactions. We denote those by $O_1, \ldots, O_l$. The environment $\mathcal{Z}$ is forced by the control program C to activate all the participants in sequence performing a "round-robin" participant execution. Each participant can ask each oracle O_k an upper bounded number of queries q_k during each round and each query has a cost c_k. The limitation in access is controlled by the control program C. The participants produce messages delivered via a "Diffuse Functionality" as in [27].

The Diffuse functionality adjusts the protocol execution in rounds and determines the communication between the honest participants and the adversary. Specifically it allows the adversary to see the messages produced by the honest participants and delay them until the end of the round. So the adversary can deliver first its messages. However at the end of each round, the Diffuse functionality delivers to all the honest participants all the messages sent from the other honest participants. Note that the Diffuse functionality gives the opportunity to the adversary to deliver first its own messages to the honest participants.

In order to model our notion of equilibrium we need to compare between two possible executions across arbitrary environments. Given this, it is important to fix the number of rounds the environment runs the protocol. To accommodate this, we will define as *r-admissible* an environment which performs the protocol a number of rounds $r = p(\kappa) \geq \kappa$, where p is a polynomial, after which it will terminate the execution. Note also that in line with [19, 20] the input of the environment will be $1^{p'(\kappa)}$, where p' is a polynomial.

The Reward and Cost Functions. We associate with a protocol Π, a *reward function* that determines the virtual rewards of each set of participants given a local view of a participant that does not belong to the coalition after the last round r of the execution. Each participant may have a different local view and as a result different conclusion regarding the rewards of other participants. Note that in a blockchain protocol this local view is reflected in the blockchain maintained by the participant. Formally: $\mathbb{E}$ is the set of all the executions of the protocol Π with respect to any adversary and environment. Note that an execution $\mathcal{E}$ is completely determined by the adversary $\mathcal{A}$, the environment $\mathcal{Z}$, the control program C and the randomness of these processes, as all the honest participants follow the protocol Π. The randomness determines the private coins of the participants, the environment, the adversary, and the oracles like the random oracle if they exist as e.g., in [27]. We use $\mathcal{E}_{\mathcal{Z},\mathcal{A}}$ to denote this random variable, where we have specified the environment and the adversary but not the randomness.[3]

The function $R_T^j : \mathbb{E} \longrightarrow \mathbb{R}$ is called the reward function and maps an execution $\mathcal{E} \in \mathbb{E}$ to the virtual rewards of a set T of participants according to the local view of a participant P_j with $P_j \in S \setminus T$ after the last round r of the execution. As an example, in the Bitcoin blockchain protocol we can consider the rewards for each participant to be the block rewards from the blocks that it has produced plus the transaction fees of the transactions included in these blocks. We define also

$$R_T^{\min}(\mathcal{E}_{\mathcal{Z},\mathcal{A}}) = \min\{R_T^j(\mathcal{E}_{\mathcal{Z},\mathcal{A}})\}_{P_j \in S \setminus T}$$

and

$$R_T^{\max}(\mathcal{E}_{\mathcal{Z},\mathcal{A}}) = \max\{R_T^j(\mathcal{E}_{\mathcal{Z},\mathcal{A}})\}_{P_j \in S \setminus T}$$

The function $C_i : \mathbb{E} \longrightarrow \mathbb{R}$ is called the *cost function* and maps an execution $\mathcal{E} \in \mathbb{E}$ to the cost a participant P_i incurred because of the queries it did to the oracles until the end of the last round r of the execution $\mathcal{E}$.

As we will explain later in our results: (i) when we want to take into account cost of mining, we assume that each participant can ask an oracle, called Random Oracle, at most q queries and incurs cost c for each query. (ii) when we want to take into account the cost of processing transactions, we assume that each participant asks an oracle, called Transaction Oracle, one query during each round and incurs cost c_{bk} for each query. In the latter case, if a coalition/pool is created, then only the *pool leader*(the participant who creates the coalition) asks the Transaction Oracle once a round and incurs c_{bk} per round. This captures the fact that the same set of transactions can be used by all members of the pool (cf. Section 4 for more details).

Remark 1. *We assume that virtual rewards and costs are directly comparable and any exchange rate between virtual rewards and cost tokens is constant and is applied directly. For instance, the participants themselves may also be acting as exchanges between virtual rewards and fiat currency or they may be delivering services that offset the costs directly.*[4] *Extending our results to a setting where a fluctuating exchange rate in the course of the execution exists between virtual rewards and cost tokens is an interesting direction for future work.*

Utility with Virtual Payoffs. We next define the (virtual) utility of a coalition of participants that are controlled by a single rational entity, the adversary. The utility may take various forms and we will consider settings where the adversary cares about its absolute rewards, its relative rewards or its absolute rewards minus its absolute cost. Other types of utility may also be defined, e.g., the adversary

[3]For simplicity we omit reference to the control program because it is the same in all the executions.

[4]This does not interfere with the fact that some participants may have diverging opinions on the rewards of another participant. The diverging opinion is based on the amount of the virtual currency recorded in their bookkeeping tables and not the exchange rate which is assumed fixed.

may want to minimize the rewards of a specific participant. We will describe the utility of a coalition controlled by a static adversary that includes the set of participants $T = \{P_{i_1}, ..., P_{i_{t'}}\} \subseteq \{P_1, ..., P_n\}$.

Definition 2.1. We define the utility function of a T-coalition in the view of a participant P_j as a function $U_T^j : \mathbb{E} \longrightarrow \mathbb{R}$ that maps an execution of $\mathbb{E}$ to a real value.

Based on the above, we define also

$$U_T^{\max}(\mathcal{E}_{\mathcal{Z},\mathcal{A}}) = \max_{P_j \in S \setminus T} \{U_T^j(\mathcal{E}_{\mathcal{Z},\mathcal{A}})\}$$

and

$$U_T^{\min}(\mathcal{E}_{\mathcal{Z},\mathcal{A}}) = \min_{P_j \in S \setminus T} \{U_T^j(\mathcal{E}_{\mathcal{Z},\mathcal{A}})\}$$

Using the reward and cost functions from the previous sections, we define below a few types of utilities that will be relevant in our analysis:

Definition 2.2. Different types of utility of a coalition T defined over an arbitrary $\mathcal{E} \in \mathbb{E}$:

- **Absolute Rewards.** $U_T^j(\mathcal{E}) = R_T^j(\mathcal{E})$,
- **Absolute Rewards minus Absolute Cost.** $U_T^j(\mathcal{E}) = R_T^j(\mathcal{E}) - \sum_{l:P_l \in T} C_l(\mathcal{E})$,
- **Relative Rewards.** $U_T^j(\mathcal{E}) = \dfrac{R_T^j(\mathcal{E})}{R_S^j(\mathcal{E})}$ if defined and 0 otherwise.
- **Relative Rewards minus Relative Cost.** $U_T^j(\mathcal{E}) = \dfrac{R_T^j(\mathcal{E})}{R_S^j(\mathcal{E})} - \dfrac{\sum_{l \in T} C_l(\mathcal{E})}{\sum_{l \in S} C_l(\mathcal{E})}$, if defined and 0 otherwise.

Coalition Safe Equilibria with Virtual Payoffs. We will examine two executions of a protocol Π with the same environment, but with different adversary and randomness. In both executions the adversary corrupts the same set of participants denoted by T with cardinality t' that is upper bounded by t. The adversary is *static* which means that during the execution it does not corrupt other participants that are not included in T. In the first execution $\mathcal{E}_{\mathcal{Z},H_T}$ the adversary runs the H_T program which means that it follows the protocol Π, i.e., plays "honestly", but it takes advantage of its network presence. Note that in our case "taking advantage of its network presence" means that the adversary delivers its messages first, when multiple competing solutions or in other words messages (such as proof of work instances) are produced during a round.[5] In the second execution $\mathcal{E}'_{\mathcal{Z},\mathcal{A}}$ the adversary is denoted by $\mathcal{A}$ and might deviate in some arbitrary way from Π. For example, in a proof of work blockchain protocol a possible deviation would be to perform selfish mining [24].

Definition 2.3. Let ϵ, ϵ' be positive constants near (or equal to) zero and r a polynomial in κ, the security parameter. The protocol is (t, ϵ, ϵ')-equilibrium with virtual payoffs (EVP) according to a utility $\{U_T^j\}_{P_j \in S \setminus T}$ when for every PPT static adversary $\mathcal{A}$ that controls an arbitrary set of $t' \leq t$ participants indexed by T and for every r-admissible environment $\mathcal{Z}$, it holds that

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

with overwhelming probability in κ. $\mathcal{E}_{\mathcal{Z},H_T}, \mathcal{E}'_{\mathcal{Z},\mathcal{A}}$ are two random variables that represent two independent executions with the same environment $\mathcal{Z}$ and adversary H_T and $\mathcal{A}$ respectively.

Remark 2. *Note that we need absolute value on the right side of the inequality because $U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T})$ can be negative when for example it is equal to the profit (absolute rewards minus absolute cost) of a participant. We use two parameters, ϵ and ϵ', to explicitly account for multiplicative and additive deviations in the utility of the diverging adversarial coalition of participants.*

Remark 3. *When the adversary selects the strategy that the participants controlled by the adversary do not ask any query and do not participate at all, then its utility is zero for all possible choices of utility from Definition 2.2. As a result if a protocol is an EVP then this implies that the utility of H_T will be not significantly smaller than 0. This parallels the participation constraint that is encountered in optimization problems in economics [32].*

The definition is generic and includes all probabilistic polynomial time (PPT) static adversaries but in our results we will consider for simplicity a *static PPT adversary with fixed cost* who decides in the beginning how many queries the participants that it controls will ask (and thus how much cost will incur). Recall that this type of cost model in the setting of proof of work blockchains is consistent with cloud mining [1]. Formally, we have the following.

Definition 2.4. A static adversary with fixed cost is an adversary that chooses in the beginning of the execution to control a set $T = \{P_{i_1}, ..., P_{i_{t'}}\} \subseteq \{P_1, ..., P_n\}$ of t' participants and it commits to the number of queries (of the available q) each participant P_{i_m}, $(m = 1, \ldots, t')$ that it controls will ask the hash function oracle *Random Oracle* during each round of the execution. This number is denoted by $q - x_m$. The number of the queries that the adversary will *not* ask the Random Oracle during a round will be denoted by x. This type of adversary can choose any strategy, but it is committed to paying during each round the cost that it chose in the beginning of the execution.

3 INCENTIVES IN BITCOIN

In this section we lay out some basic results about the Bitcoin protocol's EVP properties. We adopt the abstraction of [27] and we capture cost of mining by ascribing a cost c for each query to the Random Oracle. Each query to the Random Oracle has probability p to give a solution which is a valid block that extends the chain. We assume that each block gives a *flat reward* w to the participant who produced it. Each participant can ask the Random Oracle at most q queries per round. In this setting we provide positive results (that following Bitcoin is EVP) for utilities *absolute rewards* and *absolute rewards minus absolute cost* and negative results (that Bitcoin is not EVP for reasonable approximation parameters) for utility *relative rewards*.

In the next section we extend our setting and we take into account also the cost of processing transactions.

[5] We do not consider in this present treatment the cost of having a high presence in the network. Moreover, it is relatively easy to see that if network dominance is given at no cost, it is a rational choice for an adversary to opt for it in the Bitcoin setting since it will guarantee that more rewards will be accrued over time. We note that a similar type of reasoning was adopted also in [12] and the corresponding adversary was referred to as "front running."

Notation	
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
t'	number of participants controlled by the adversary
t	upper bound of t'
r	round after which an execution terminates
n	number of participants
w	flat reward per block (Bitcoin)
w_f	flat reward per fruit (Fruitchain [53])
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
c_{bk}	cost of processing transactions per round by a single node
c	cost of a single query to the Random Oracle.

The expected number of solutions per round by all participants is denoted by s. We note that s is assumed to be close to zero (e.g., in [27] it is noted that $s \approx 0.03$ for the Bitcoin protocol parametrization and network conditions).

In the statements below, recall t' is the number of the participants controlled by the adversary, t is the upper bound on t', S is the set of all the participants and T the set controlled by the adversary. The results are as follows (for the proofs see the full version of this paper in [39]).

Absolute rewards: When the utility is based on absolute rewards (cf. Def.2.2), then Bitcoin with fixed target is EVP against a coalition that includes even up to *all but one* of the participants. This is in agreement with the result of [34]. The intuition behind this result is that if the utility only depends on how many blocks are produced, then there is no incentive to deviate from the protocol — e.g., by creating forks or by keeping blocks private. The reason is that if one deviates from the protocol then it increases the possibility that its blocks will not be included in the public ledger compared to following the protocol. Moreover, the number of the blocks the adversary produces during a round depends only on p, q, t and not on which chain the adversary extends.

Theorem 3.1. *For any $\delta_1 \in (0, 0.25)$ such that $4 \cdot \delta_1 \cdot (1+s)+s < 1$, where s the expected number of solutions per round, Bitcoin with fixed target in a synchronous setting where the reward of each block is a constant, is $(n-1, 4 \cdot \delta_1 \cdot (1+s)+s, 0)$-EVP according to the utility function absolute rewards (Def. 2.2).*

Note that the better synchronization we have (the fewer expected number of solutions per round s) then the better EVP [6] we have (the lower $4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s$ is). Recall $4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s$ is related to how much the adversary can gain if it deviates. Moreover, in the theorem we allow the adversary to control all but one of the participants (and not all) because we want at least one honest local chain according to which we can determine the rewards of the adversary.

[6] By "better EVP" we mean that the actual values of ϵ, ϵ' are smaller.

We extend the above result also in the setting where the block reward changes after a period of $l \cdot \kappa$ or more rounds where l a positive constant and κ the security parameter during the execution.

Theorem 3.2. *Supposing that (i) the block reward changes every at least $l \cdot \kappa$ rounds where l 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 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 (def.2.2).*

Note that in the analysis above, we assume throughout that the *target* used in the proof of work function remains *fixed* as in [27]. It is easy to see that if this does not hold, then the adversary using selfish mining [24] can cause the protocol to adopt a target that becomes greater than what is supposed to be, and thus the difficulty in mining a block will decrease, as the total computational power would appear smaller than it really is. In this case, the adversary can produce blocks faster and as such it can magnify its rewards resulting in a negative result in terms of EVP (see also [31]). It is an easy corollary that the protocol will not be an EVP in this case.

Absolute rewards minus absolute cost: When the utility is based on absolute rewards minus absolute cost then the Bitcoin protocol with fixed target is EVP against a coalition that controls even up to all but one of the participants, assuming the cost of each query c is small enough compared to the block reward w. Again the better synchronization we have, the better EVP we have.

Theorem 3.3. *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 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 (Def. 2.2).*

Remark 4. *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 of*

each block is high enough to compensate the miners for 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 as proved in [25].

Note that the smaller the cost of each query is, the better EVP we have (because we can select smaller ϕ such that $(4 \cdot \delta_1 + (1 + 4 \cdot \delta_1) \cdot s)/(1 - \phi)$ is smaller).

We extend the above result also to the case when the block reward changes every at least $l \cdot \kappa$ rounds where l a positive constant and κ the security parameter.

Theorem 3.4. *Assume that (i) the block reward changes every at least $l \cdot \kappa$ rounds where l 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 player. 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 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 (Def. 2.2).*

Relative rewards: When the utility is based on relative rewards, i.e., the ratio of rewards of the strategic coalition of the adversary over the total rewards of all the participants, Bitcoin with fixed target cannot be an EVP with small ϵ, ϵ'. The core idea is to use the selfish mining strategy [22, 24, 29, 51, 57] to construct an attack that invalidates the equilibrium property. This kind of attack was used also in [27] as argument for the tightness of "chain quality" (chain quality refers to the fraction of the blocks in the public ledger that belong to the adversary). Without loss of generality, we will assume that the reward of each block is the same and equal to w (the negative result carries trivially to the general case).

The result is in agreement with [13, 24] and an argument regarding incentive compatibility of Bitcoin presented in [53]. However it seems to contradict the result from [38], which shows that in a "strategic-release game" that describes Bitcoin, honest strategy is Nash equilibrium when the adversary controls a small coalition. This difference arises because that model assumes that all honest miners act as a single miner which implies that when an honest participant produces a block, all the other honest participants adopt this block, something that does not happen in our setting where the adversary is assumed to have network dominance. Note that in [27, 53] and in our case the adversary has the advantage that it can always deliver its block first and the honest participants adopt the first block they receive. As a result, the blocks of the adversary never become dropped in a case when both the adversary and an honest participant produce a block during a round.

Theorem 3.5. *Let $t \in \{1, ..., n - 1\}$ and $t' < \min\{n/2, t + 1\}$. Then for any $\epsilon + \epsilon' < \frac{t'}{n - t'} \cdot (1 - \delta') - \frac{t'}{n} \cdot (1 + \delta'') \cdot (1 + s)$, for some δ', δ'' close to zero, where s the expected number of solutions per round, following Bitcoin with fixed target in a synchronous setting is* not *a (t, ϵ, ϵ')-EVP according to the utility function relative rewards (Def. 2.2).*

Now we will describe what happens when we remove the assumption that each block gives a fixed flat reward.

3.1 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 [12, 27] 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 H_T. Some examples are some attacks described in [17, 45]. 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 [12] that describes some distributions of inputs which make Bitcoin not incentive compatible. Another related work that studies incentives in proof of work blockchain protocols (and specifically in Ethereum [2]) when we take into account transaction rewards is [56].

4 MINING POOLS AND CENTRALIZATION

In this section we extend our setting and we take into account also an additional cost to prepare the contents of a block. Recall that in Bitcoin transactions are collected, verified and organized in a list, and subsequently processed into an authenticated data structure (a Merkle tree). Subsequently the root of the tree together with some additional information forms the proof of work instance that is used to repeatedly query the Random Oracle.

We capture this as a tangible cost in the following way: (i) each participant asks the *Transaction Oracle* one query per round and incurs cost c_{bk} per round for processing transactions. Note that c_{bk} will only be charged once in a round, something that reflects the fact that one need not repeat the block content structuring every time a a proof of work hash query (query to the Random Oracle) is attempted (on the other hand, we do charge c_{bk} once per round, reflecting the fact that new transactions will be continuously processed). (ii) when a pool T is created, then *only* the pool leader of this coalition asks the Transaction Oracle and incurs c_{bk} per round. The latter captures the fact that when a set of transactions has been processed by the pool leader of a pool, then it can be used by all the members of the pool. Note that in the case of Random Oracle queries, this does not happen because joining a pool does not reduce the queries it has to pose and the cost of the electricity these queries incur.

We next introduce the following strategy which deviates from the Bitcoin protocol.

Definition 4.1. For some $P_i \in S$, the protocol strategy M_i is divided in two protocols: (i) participant P_i is the mining *pool leader*, collects transactions, processes them and then sends the POW

instance x_p to all participants to work on (including itself), (ii) participant P_j for all j, receives the instance x_p and executes the main proof of work main loop. Let $\tilde{c}_{\text{bk}}$ be c_{bk} multiplied by the number of rounds a block needed to be produced. We will assume that the reward w per block is so high that with overwhelming probability for every block it holds $w > \tilde{c}_{\text{bk}}$.

When a block is produced and $\tilde{c}_{\text{bk}}$ is not higher than block's reward w (which happens with overwhelming probability) then P_i issues payments equally to all participants (recall we are in the flat model where all participants have the same computational power) while retaining for itself a reward $w/n + \tilde{c}_{\text{bk}} - \tilde{c}_{\text{bk}}/n$. If $\tilde{c}_{\text{bk}} > w$ then P_i retains w for itself.

If after a successful round no payment to some participant $P_j \neq P_i$ is provided by P_i, participant P_j switches to regular mining, i.e., operates identically to the standard Bitcoin protocol. In addition, if a participant P_j produces a block that does not belong to the pool (uses a different POW instance x_p from what suggested by P_i) then it is excluded from the pool. Note that with this protocol allocation *all* blocks in the blockchain may be produced by participant P_i.

At this point we prove that following the above "single-mining-pool" strategy M_i for any $P_i \in S$ is an EVP. This holds even when the cost to prepare the content of a block is much lower compared to the cost of mining. The intuition behind the following theorem is that the participants (i) have the same expected rewards when they follow M_i (and mine for the pool) and when they mine alone. (ii) when they mine for the pool, then they share the cost of processing transactions until the last block of the pool is produced (after that it is paid exclusively by the pool leader). To be more precise, c_{bk} is paid only by the pool leader during each round, but when a block is produced, the pool leader retains the cost that it incurred before sharing block's reward. So given that w is higher than the cost of processing transactions $\tilde{c}_{\text{bk}}$ with overwhelming probability, this has the same effect as all the members share the cost.

Theorem 4.2. *Let us assume that $c_{\text{bk}} < c \cdot q$ and that there exists ϕ such that $c < p \cdot w \cdot \phi$. Then for every δ, l such that $l \in (0, \frac{r}{\kappa})$, $\delta \in (0, \frac{l \cdot \kappa}{2 \cdot (r - l \cdot \kappa/2)})$ and $c < p \cdot w \cdot \phi \cdot (1 - \delta)$ following "single-mining-pool" strategy M_i for any $P_i \in S$ is an $(n-1, \frac{4 \cdot \delta}{1 - 2 \cdot \phi}, (l \cdot \kappa \cdot s \cdot (1+\delta) \cdot \frac{1}{4} + 1) \cdot w - c_{\text{bk}})$-EVP according to the utility function absolute rewards minus absolute cost (Def. 2.2).*

We note that the multiplicative parameter $\epsilon = \frac{4 \cdot \delta}{1 - 2 \cdot \phi}$ is related to how much the adversary can gain by dissolving its pool at the first round and following standard Bitcoin. The additive parameter $\epsilon' = (l \cdot \kappa \cdot s \cdot (1+\delta) \cdot \frac{1}{4} + 1) \cdot w - c_{\text{bk}}$ is related to how much the adversary can gain by betraying its pool during the execution and either not issuing payments or producing a block that does not belong to the pool.

Proof. We consider two cases w.r.t. the strategy M_i. In the first, the adversarial coalition T with $t' \in \{1, \ldots n-1\}$ participants does not include participant P_i. In the second, the adversarial coalition includes participant P_i. For simplicity, we will assume that if the pool leader dissolves the pool, then it will not create it again, and also if a member is excluded from the pool or abandons the pool, then it will not join it again.

The adversarial coalition includes P_i. In this case the alternative strategies of the coalition are the following: (i) the coalition dissolves its pool at the first round and mines on its own. Note that it can follow an arbitrary strategy and it is not restricted to follow the Bitcoin protocol when it dissolves its pool. The coalition who follows this strategy will be denoted by A_1. (ii) the coalition follows M_i until a round r^* that is successful for the pool at which the adversarial coalition "betrays" the pool by not issuing the payments of the honest participants. At this point the pool is dissolved because the honest participants abandon the pool according to M_i. The coalition who follows this strategy will be denoted by A_{2,r^*}. Note that the case where the coalition does not issue the payment of a subset S_0 of the honest participants is captured by the case where the adversarial coalition T includes all the participants except S_0.

Observe that when all the participants follow M_i and mine for the pool, then they have the same local chain.

Let $\phi \in (0,1)$ such that $c < p \cdot w \cdot \phi$, $l \in (0, \frac{r}{\kappa})$ and $\delta \in (0, \frac{l \cdot \kappa}{2 \cdot (r - l \cdot \kappa/2)})$ such that $c < p \cdot w \cdot \phi \cdot (1 - \delta)$. Recall that r is the number of rounds the environment runs the execution.

Then we want to prove that for any r-admissible environment $\mathcal{Z}$ and adversary A that follows a strategy described in (i),(ii) it holds

$$U_T^{\max}(\mathcal{E}'_{\mathcal{Z},\mathcal{A}}) \leq U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) \cdot (1 + 4 \cdot \delta_1 \cdot \frac{1}{1 - 2 \cdot \phi}) + l \cdot \kappa \cdot s \cdot (1+\delta) \cdot w \cdot \frac{1}{4} + w$$

with overwhelming probability in the security parameter κ. Recall that s is the expected number of blocks produced by all participants during a round and is very close to zero in the synchronous model. In this case H_T is the coalition T that follows M_i. Let $Y_{\mathcal{Z},\mathcal{A}}$ be the above condition.

Now we will prove that for every adversary A_1 that follows (i) $Y_{\mathcal{Z},A_1}$ is satisfied. If we assume that $c_{\text{bk}} < c \cdot q$ then we can prove $Y_{\mathcal{Z},A_1}$ for every adversary A_1 in the same way as the proof of Theorem 6.2 (we omit the details because we will present this proof in the next section). The approximation multiplicative parameter ϵ in this case is $4 \cdot \delta_1 \cdot \frac{1}{1 - 2 \cdot \phi}$ instead of $4 \cdot \delta_1 \cdot \frac{1}{1 - \phi}$ that is the approximation parameter in Theorem 6.2. This happens because in this case the utility of the adversary includes the cost of verifying transactions per round.

The intuition why this proof holds is the following: when the coalition follows M_i, it will take in expectation the same rewards from the blocks it will produce as mining on its own following the Bitcoin protocol, and it will incur at most the same cost. Observe that when it follows M_i, the cost of verifying transactions will be shared among all the members of the pool and it will not get paid exclusively by the coalition as happens when it mines on its own. Also, as it holds $c < p \cdot w \cdot \phi \cdot (1 - \delta)$, when the coalition mines on its own, the most profitable strategy is to ask all the hashing queries and follow the Bitcoin protocol (except the following: if it produces more than one block during a round, it sends all the blocks to the Diffuse Functionality ,as happens with fruits in the Fruitchain protocol).

Note that even if $c_{\text{bk}} < c \cdot q$ does not hold, then again we can prove a similar result where the multiplicative approximation parameter $\epsilon = 0$ and the additive approximation parameter $\epsilon' = 2 \cdot \delta \cdot s \cdot r \cdot w$.

Now will prove that for every adversary A_{2,r^*} that follows (ii) and betrays its pool by not issuing the condition payments of the honest parties at round r^* the condition $Y_{\mathcal{Z},A_{2,r^*}}$ is satisfied.

Until the beginning of round r^* the adversary A_{2,r^*} follows the same strategy as H_T (following M_i) and also all the parties have the same local chain. Thus until the beginning of round r^*, for the same randomness, A_{2,r^*} and H_T have the same profit according to any honest participant's local chain.

Let $\mathcal{E}'_{\mathcal{Z},A_{2,r^*},r^*}$ be the execution $\mathcal{E}'_{\mathcal{Z},A_{2,r^*}}$ restricted to rounds $r^*,\dots r$ and $\mathcal{E}'_{\mathcal{Z},A_{2,r^*},-r^*}$ be the execution $\mathcal{E}'_{\mathcal{Z},A_{2,r^*}}$ restricted to rounds $1,\dots r^*-1$.

By the above observation we can conclude that:

(1) $U_T^{\max}(\mathcal{E}'_{\mathcal{Z},A_{2,r^*}}) = X + X_1(r^*)$ where we have $X_1(r^*) = U_T^{\max}(\mathcal{E}'_{\mathcal{Z},A_{2,r^*},r^*})$ and
$$X = U_T^{\max}(\mathcal{E}'_{\mathcal{Z},A_{2,r^*},-r^*}) = U_T^{\min}(\mathcal{E}_{\mathcal{Z},\mathcal{H}_T,-r^*})$$

(2) $U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T}) = X + X_2(r^*)$ where $X_2(r^*) = U_T^{\min}(\mathcal{E}_{\mathcal{Z},H_T},r^*)$

At this point we treat the random variables as functions that have as input the random coins of the execution and output the maximum (among honest participant's local chains) utility of A_{2,r^*} and minimum utility of H_T respectively. Recall that when the coalition follows M_i until round r^*, then at the beginning of round r^* all the participants have the same local chains. In addition the coalition can deliver its block first, so the maximum and minimum utility for rounds $r^*,\dots r$ are the same.

Thus if we prove that the event $X_1(r^*) - X_2(r^*) \le l\cdot\kappa\cdot s\cdot(1+\delta)\cdot w\cdot\frac{1}{4} + w$ denoted by $D(r^*)$ happens with overwhelming probability then $Y_{\mathcal{Z},A_{2,r^*}}$ is satisfied.

- It holds with overwhelming probability that $X_1(r^*) = w\cdot(y(r^*)+1) - (c_{\mathsf{bk}} + c\cdot q\cdot t')\cdot(r-r^*+1)$ where $y(r^*)$ is the number of the successful queries for T after round r^*.
- It holds with overwhelming probability that $X_2(r^*) \ge c_{\mathsf{bk}} + w\cdot y'(r^*)\cdot\frac{t'}{n} - c_{\mathsf{bk}}\cdot(r-r^*+1) - c\cdot q\cdot t'\cdot(r-r^*+1)$ where $y'(r^*)$ are the number of successful queries for all the members of the pool after round r^*. Note that $y'(r^*) = x(r^*) + y(r^*)$ where $x(r^*)$ are the number of successful queries for the honest parties after round r^*. Recall that when a block is produced by the pool, then according to M_i, the coalition's reward from this block is $\tilde{c}_{\mathsf{bk}}$ (which is equal to c_{bk} multiplied by the number of rounds the block needed to be produced) plus $(w-\tilde{c}_{\mathsf{bk}})\cdot\frac{t'}{n}$. We have assumed that w is higher than $\tilde{c}_{\mathsf{bk}}$ with overwhelming probability. Thus the coalition's rewards from the block at round r^* are at least $\tilde{c}_{\mathsf{bk}} \ge c_{\mathsf{bk}}$ with overwhelming probability.

As a result with overwhelming probability the following event holds

$$\begin{aligned} X_1(r^*) - X_2(r^*) &\le (y(r^*) - y'(r^*)\frac{t'}{n})\cdot w + w - c_{\mathsf{bk}} \\ &= (y(r^*)\cdot\frac{n-t'}{n} - x(r^*)\cdot\frac{t'}{n})\cdot w + w - c_{\mathsf{bk}} \end{aligned}$$

Observe that when the coalition follows M_i after r^*, it takes a fraction t'/n from every block produced by any participant. On the other hand, when it mines alone, it gets a reward only from the blocks it produces but when it produces a block retains its whole reward. Note that the above is an upper bound because when the coalition follows M_i, then it will share the cost of verifying the transactions.

If we show that for every $r^* \in \{1,\dots,r\}$ the following event denoted by $V(r^*)$ happens with overwhelming probability then we conclude that for every $r^* \in \{1,\dots,r\}$ the event $D(r^*)$ happens with overwhelming probability :

$$y(r^*)\cdot\frac{n-t'}{n} - x(r^*)\cdot\frac{t'}{n} \le l\cdot\kappa\cdot s\cdot(1+\delta)\cdot\frac{1}{4}$$

.

- When $r^* > r - l\cdot\kappa - 1$ then $y(r^*) \le y(r - l\cdot\kappa - 1)$ with probability 1 and by Chernoff bound we have $y(r-l\cdot\kappa-1) \le l\cdot\kappa\cdot p\cdot q\cdot t'\cdot(1+\delta)$ with overwhelming probability in $l\cdot\kappa$. Thus the event $y(r^*)\cdot\frac{n-t'}{n} \le l\cdot\kappa\cdot s\cdot(1+\delta)\cdot\frac{1}{4}$ happens with overwhelming probability. Note that $t'\cdot(n-t')$ is maximized for $t' = n/2$.
- When $r^* \le r - l\cdot\kappa$ then by Chernoff bound it holds with overwhelming probability in $r - r^*$ and thus in κ

$$\begin{aligned} &y(r^*)\cdot\frac{n-t'}{n} - x'(r^*)\cdot\frac{t'}{n} \le (r-r^*)\cdot p\cdot q\cdot t'\cdot(1+\delta)\cdot\frac{n-t'}{n} \\ &- (r-r^*)\cdot p\cdot q\cdot(n-t')\cdot(1-\delta)\frac{t'}{n} \\ &\le 2\cdot\delta\cdot s\cdot(r-r^*)\cdot\frac{1}{4} \\ &\le l\cdot\kappa\cdot s\cdot(1+\delta)\cdot\frac{1}{4} \end{aligned}$$

Recall that $\delta \in (0, \frac{l\cdot\kappa}{2\cdot(r-l\cdot\kappa/2)})$.

The adversarial coalition does not include P_i. This case is captured by the previous case (where the adversarial coalition includes P_i) because (i) when a member produces a block that does not belong to the pool, it is excluded from the pool (ii) the pool leader does not earn extra rewards compared to the members. Moreover, with overwhelming probability the profit of the members is (a) the same as pool leader's profit, if a block is produced at the last round of the execution (b) higher than pool leader's profit, when the last block of the pool was produced before the last round. This happens because until the round the last block is produced, the pool leader pays c_{bk} (because it asks the Transaction Oracle), but when a block is produced, before it issues the payments, it retains an amount equal to the cost of processing transactions it has incurred (so it is the same as not paying anything). After the last block, it pays c_{bk} for each round and it cannot retain it as no block is produced (compared to the members that do not pay c_{bk}).

□

5 WEAK FAIRNESS AND EVP

In this section we describe a property, called "(t,δ)-weak fairness", which is sufficient for proving that a protocol is EVP when the utility is based on relative rewards (cf. Def.2.2). This property can aid in the design of EVP protocols.

A protocol will satisfy "(t,δ)-weak fairness" property when with overwhelming probability the following hold: firstly when the adversary (which controls at most t participants) deviates, then the fraction of the rewards that the set of all the honest participants gets is at least $(1-\delta)$ multiplied by its relative cost and secondly when the adversary is H_T, which means that it follows the protocol,

any set of participants gets at least $(1-\delta)$ multiplied by its relative cost.

Definition 5.1. 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 $j : P_j \in S \setminus T$, where S the set of all the participants, we have with overwhelming probability in the security parameter κ:

- $R^j_{S\setminus T}(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \geq (1-\delta)\cdot \frac{\sum_{l:P_l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})}\cdot R^j_S(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})$
- for any subset $S_H \subseteq S$ it holds $R^j_{S_H}(\mathcal{E}_{\mathcal{Z},H_T}) \geq (1-\delta)\cdot \frac{\sum_{l:P_l\in S_H} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})}\cdot R^j_S(\mathcal{E}_{\mathcal{Z},H_T})$ where $\delta \in [0,1)$.

Note that $\sum_{l:P_l\in S_H} C_l(\mathcal{E}_{\mathcal{Z},H_T})/\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})$ represents the computational power of S_H[7], because honest participants and H_T ask all the queries during each round. In addition it holds

$$\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T}) \neq 0$$

as the execution lasts at least one round. We do not divide with $R^j_S(\mathcal{E}_{\mathcal{Z},H_T})$ as we do not exclude the case that is equal to zero.

Our property is weaker than (T,δ)-*approximate fairness w.r.t.* ρ *attackers* defined in [53] and *ideal chain quality* defined in [27].

The property (T,δ)-*approximate fairness w.r.t.* ρ *attackers* defined in [53] says that in any sufficient long window of the chain with T blocks, any set of honest participants with computational power ϕ will get 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 [27] says that any coalition of participants (regardless the mining strategy they follow) will get a percentage of blocks in the blockchain that is proportional to their collective computational power.

Our property is weaker than (T_0,δ)-*approximate fairness w.r.t.* t/n *attackers* (n is the number of all the participants)[8] defined in [53] and *ideal chain quality* in [27] in the sense that when the adversary deviates from the protocol we demand that only the whole set of the honest participants gets a fraction of rewards at least $(1-\delta)$ multiplied by its relative cost, not all the subsets of the honest participants. In the same way our definition is also weaker than *race-free property* defined in [14][9].

According to the following theorem when a protocol satisfies the (t,δ)-weak fairness property and the total rewards are greater than zero with overwhelming probability, then following the protocol is EVP under an adversary that controls at most t participants. This theorem will be also used in order to prove that the Fruitchain protocol [53] is EVP when the utility is based on relative rewards.

Theorem 5.2. *When a protocol satisfies (t,δ)-weak fairness and in addition for any $j : P_j \in S \setminus T$, 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 $R^j_S(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) > 0$ with overwhelming probability in the security parameter κ, then following the protocol is $(t,0,\delta)$-EVP according to the utility function relative rewards (def.2.2).*

Proof. We choose an arbitrary r-admissible environment $\mathcal{Z}$, where κ is the security parameter and an arbitrary adversary $\mathcal{A}$ static that is PPT and it controls a set T that it includes $t' \leq t$ participants. We will 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 will prove that with overwhelming probability in the security parameter for any $j : P_j$ honest we have:

$$U^j_T(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) = \frac{R^j_T(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})}{R^j_S(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})} \tag{1}$$

$$\leq \frac{\sum_{l:P_l\in T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} + \delta\cdot\frac{\sum_{l:P_l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} \tag{2}$$

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

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

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

$$R^j_T(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \leq R^j_S(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})\cdot\left(1-(1-\delta)\cdot\frac{\sum_{l:P_l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})}\right) \Rightarrow$$

$$R^j_T(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) \leq R^j_S(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})\cdot\left(\frac{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} - (1-\delta)\cdot\frac{\sum_{l:P_l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})}\right) \Rightarrow$$

$$\frac{R^j_T(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})}{R^j_S(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})} \leq \frac{\sum_{l:P_l\in T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} + \delta\cdot\frac{\sum_{l:P_l\in S\setminus T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})}$$

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

$$U^j_T(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) = \frac{R^j_T(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})}{R^j_S(\mathcal{E'}_{\mathcal{Z},\mathcal{A}})} \tag{3}$$

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

$$U^{min}_T(\mathcal{E}_{\mathcal{Z},H_T}) \geq (1-\delta)\cdot\frac{\sum_{l:P_l\in T} C_l(\mathcal{E}_{\mathcal{Z},H_T})}{\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T})} \tag{4}$$

[7] $\sum_{l:P_l\in S_H} C_l(\mathcal{E}_{\mathcal{Z},H_T})/\sum_{l:P_l\in S} C_l(\mathcal{E}_{\mathcal{Z},H_T}) = (c\cdot q\cdot r\cdot t_H)/(c\cdot q\cdot r\cdot n)$ where t_H the number of participants of S_H

[8] To be precise it is weaker than fairness under the restriction that the environment performs the protocol so many rounds that with overwhelming probability any honest participant has a local chain of length at least T_0. Note that this happens because in our definition we have not used T_0 as parameter.

[9] Note that when a cryptocurrency is pseudonymous and not anonymous then it is difficult to secure that every subset of honest participants will take the appropriate percentage of the blocks, because maybe it is the case where the adversary cannot decrease much the percentage of the blocks that belongs to the whole set of the honest participants, but it can act against a specific participant with some characteristics revealed from the graph of the transactions. For example there are some works that analyze the statistical properties of the Bitcoin transaction graph and describe identification attacks in Bitcoin, [49, 55]

By equations (1), (4) we have that with overwhelming probability in the security parameter

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

□

6 INCENTIVES IN THE FRUITCHAIN PROTOCOL

Abstraction of the Fruitchain Protocol. In the Fruitchain protocol [53] the participants use a hash function modelled as a random oracle. The number of the queries to the random oracle by each participant during a round is bounded by q. Let the total number of the participants be n. Each query to the random oracle can give with probability p a block and with probability p_f a fruit, where p_f is assumed to be greater than p. This is achieved via the 2-for-1 POW technique of [27]. At the beginning of each round, when the honest participants are activated, they "receive" the fruits and the blocks from the Diffuse Functionality, they choose the chain that they will try to extend and they include in the block they try to produce "a fingerprint" of all the "recent" fruits (as defined in [53]) that have not been included in the blockchain yet. Then they ask the random oracle q queries. When an honest participant finds a fruit or a block, it gives it to the Diffuse Functionality and it continues asking the remaining queries. Even if it finds more than one fruit during a round, it gives all the fruits to the Diffuse Functionality. The adversary is activated at the end and it can ask $t' \cdot q$ queries, where t' is the number of the participants that it controls.

We consider that the rewards come only from the fruit[10] and the difficulty in mining a block is fixed. In our case each query to the random oracle has a cost c. In the proofs we will assume that the adversary is static, the model is synchronous and the Diffuse Functionality works as [27], and each fruit gives reward equal to w_f.

Relative rewards: According to the following theorem if the adversary controls fewer than half of the participants and wants to maximize its relative rewards which means that its utility is based on relative rewards (Def. 2.2), then following the Fruitchain protocol is EVP. This theorem allows us to understand in a formal way how mining simultaneously fruits and blocks can eliminate the impact of selfish mining [24] on the incentive compatibility of the protocol. We note that the core advantage stems from the 2-for-1 POW technique used for simultaneous mining which was initially proposed for the mitigation of selfish mining in [27] in the context of achieving Byzantine agreement for honest majority and later was adapted in [53] for a similar purpose in the context of fair blockchains.

[10]Note that in the Fruitchain protocol [53] the fairness property holds for the fruits; actual blocks are possibly still vulnerable to selfish mining attacks [24]. So if we consider that also the blocks give a flat reward then we cannot use the fairness property proved in [53].

Theorem 6.1. *Let $\delta \in (0,1)$ 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 (Def. 2.2), under an r-admissible environment where $r \geq T_0/(p_f\cdot(\frac{n}{2}+1)\cdot(1-\delta)\cdot q)$.*

The proof uses Chernoff bound and Theorem 5.2. Note that for any $\delta \in (0,1)$ and appropriate T_0 the Fruitchain protocol satisfies (T_0,δ)-approximate fairness property (Subsection 4.2 in [53])[11].

Proof. As 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 [53] when $r \geq \frac{T_0}{p_f\cdot(\frac{n}{2}+1)\cdot(1-\delta)\cdot q}$ and 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 $j: P_j$ 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 $R_S^j(\mathcal{E'}_{\mathcal{Z},\mathcal{A}}) > 0$. So by Theorem 5.2 we have that the Fruitchain protocol is $(n/2-1,0,\delta)$-EVP under an r-admissible environment where $r \geq \frac{T_0}{p_f\cdot(\frac{n}{2}+1)\cdot(1-\delta)\cdot q}$. □

Remark 5. *The above theorem holds when we take into account also the transaction fees from each fruit and at the end of the execution we distribute evenly the total rewards among the miners of the fruits (as assumed in [53][12]).*

Absolute rewards minus absolute cost: We will prove that the Fruitchain [53] protocol in a synchronous setting is EVP according to utility based on absolute rewards minus absolute cost (Def. 2.2) if the adversary controls all but one participants, when the cost of each query c is small enough compared to the reward of each fruit w_f. Note that the smaller the cost of each query is, the better EVP we have.[13]

The intuition behind the proof is that (i) the rewards come from the fruits that are produced by mining and (ii) the total number of the fruits the adversary can produce is bounded (with overwhelming probability) whatever strategy it follows. So if the adversary can have this number of fruits even if it follows the protocol, it has no reason to deviate.

Theorem 6.2. *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 (Def. 2.2).*

[11]In [53] the number of queries q each participant can ask during each round is 1.

[12]To be precise in [53] the rewards of each fruit are shared evenly among the miners of the fruits included in a long enough preceding part of the chain.

[13]Note that here synchronization does not affect how good the EVP is in contrast to our theorems regarding Bitcoin with fixed target. This happens because when honest participants find more than one fruit during a round, all of them can be included in the chain eventually

Proof. In this setting the adversary again is PPT, static with fixed cost, it controls a set of participants $T = \{P_{i_1}, ..., P_{i_{t'}}\} \subseteq \{P_1, ..., P_n\} = S$ and chooses in the beginning the number x_m of the questions that each participant controlled by the adversary P_{i_m} will not ask during each round of the execution.

Let an arbitrary $\delta_1 \in (0, 0.25)$ 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 $\mathcal{Z}$, 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 will 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:

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

with overwhelming probability in κ.

The above equation is proved by Chernoff bound and taking into account that all the fruits produced by T will be 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

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

with overwhelming probability in κ.

As a result

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

with overwhelming probability in κ. □

Remark 6. *The assumption that there exists $\phi \in (0,1)$ such that $c < p_f \cdot w_f \cdot \phi$ means that the reward of each block is high enough to compensate the miners for the cost of the mining. Finally note that trivially if we consider that $c = 0$ then the assumption of the above theorem holds for ϕ close to zero and the utility is just absolute rewards (Def. 2.2).*

7 CONCLUSION

In this work we introduced a new concept of approximate equilibrium, that is suited for the setting where participants in a distributed system seek to offset costs with *virtual payoffs*. We showed a number of results for the Bitcoin and Fruitchain blockchain protocols establishing EVP strategies for various types of utilities. We examined two major classes of utility functions, absolute and relative rewards and we established both positive and negative results for these blockchain protocols. Among others, we showed that the Bitcoin protocol admits a strategy as an EVP that fully centralizes all miners to a single mining pool.

A number of interesting directions are motivated by the present work. The first one is devising protocols that are EVPs against coalitions who adaptively fluctuate their mining resources, while the protocol itself adjusts mining difficulty to accomodate an ever fluctuating population of participants. A second one is to investigate suitable modifications for the Bitcoin protocol that prevent centralization to a single mining pool as an EVP strategy.

ACKNOWLEDGMENTS

The research was partially supported by H2020 project PRIVILEDGE #780477 and was conducted at the Blockchain Technology Laboratory at the University of Edinburgh which was established by funding of IOHK.

8 APPENDICES

A CHERNOFF BOUNDS

Let $X_i : i \in \{1, ..., n\}$ be mutually independent Boolean random variables and $\forall i Pr(X_i = 1) = p$. 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

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

B NEGLIGIBLE FUNCTIONS AND PROBABILITY EVENTS

Definition B.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 B.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.

REFERENCES

[1] [n.d.]. Cloud Mining. https://en.bitcoin.it/wiki/Cloud_mining.
[2] [n.d.]. Ethereum. https://ethereum.org/en/.
[3] [n.d.]. Hash function. https://en.wikipedia.org/wiki/Hash_function.
[4] 2007. *Algorithmic Game Theory.* Cambridge University Press. https://doi.org/10.1017/CBO9780511800481 Nisan, N., Roughgarden, T., Tardos, E., & Vazirani, V. (Eds.). doi:10.1017/CBO9780511800481.
[5] Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. 2006. 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* (Denver, Colorado, USA) *(PODC '06).* ACM, New York, NY, USA, 53–62. https://doi.org/10.1145/1146381.1146393

[6] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. 2016. Solidus: An Incentive-compatible Cryptocurrency Based on Permissionless Byzantine Consensus. *CoRR* abs/1612.02916 (2016). arXiv:1612.02916 http://arxiv.org/abs/1612.02916

[7] Amitanand S. Aiyer, Lorenzo Alvisi, Allen Clement, Mike Dahlin, Jean-Philippe Martin, and Carl Porth. 2005. BAR Fault Tolerance for Cooperative Services. In *Proceedings of the Twentieth ACM Symposium on Operating Systems Principles* (Brighton, United Kingdom) *(SOSP '05)*. ACM, New York, NY, USA, 45–58. https://doi.org/10.1145/1095810.1095816

[8] Nick Arnosti and S. Matthew Weinberg. 2018. Bitcoin: A Natural Oligopoly. *CoRR* abs/1811.08572 (2018). arXiv:1811.08572 http://arxiv.org/abs/1811.08572

[9] Robert J. Aumann. 1959. *Acceptable Points in General Cooperative n-Person Games. Contributions to the Theory of Games (AM-40)*. Vol. 4. Albert William Tucker, Robert Duncan Luce, Princeton: Princeton University Press, 287–324. Book DOI: https://doi.org/10.1515/9781400882168.

[10] Moshe Babaioff, Shahar Dobzinski, Sigal Oren, and Aviv Zohar. 2012. On Bitcoin and Red Balloons. In *Proceedings of the 13th ACM Conference on Electronic Commerce* (Valencia, Spain) *(EC '12)*. ACM, New York, NY, USA, 56–73. https://doi.org/10.1145/2229012.2229022

[11] Adam Back. 1997. Hashcash. http://www.cypherspace.org/hashcash.

[12] Christian Badertscher, Juan Garay, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. 2018. But Why Does It Work? A Rational Protocol Design Treatment of Bitcoin. In *Advances in Cryptology – EUROCRYPT 2018*, Jesper Buus Nielsen and Vincent Rijmen (Eds.). Springer International Publishing, Cham, 34–65.

[13] Suguman Bansal. 2016. *Reasoning about incentive compatibility.* POPL 2016 Student Research Competition.

[14] Iddo Bentov, Pavel Hubácek, Tal Moran, and Asaf Nadler. 2017. Tortoise and Hares Consensus: the Meshcash Framework for Incentive-Compatible, Scalable Cryptocurrencies. *IACR Cryptology ePrint Archive* 2017 (2017), 300. http://eprint.iacr.org/2017/300

[15] B.Douglas Bernheim, Bezalel Peleg, and Michael D Whinston. 1987. Coalition-Proof Nash Equilibria I. Concepts. *Journal of Economic Theory* 42, 1 (1987), 1 – 12. https://doi.org/10.1016/0022-0531(87)90099-8

[16] Bruno Biais, Christophe Bisiere, Matthieu Bouvard, and Catherine Casamatta. 2018. *The Blockchain Folk Theorem.* Swiss Finance Institute Research Paper No. 17-75.

[17] Joseph Bonneau. 2016. Why Buy When You Can Rent? - Bribery Attacks on Bitcoin-Style Consensus. In *Financial Cryptography and Data Security - FC 2016 International Workshops, BITCOIN, VOTING, and WAHC, Christ Church, Barbados, February 26, 2016, Revised Selected Papers (Lecture Notes in Computer Science, Vol. 9604)*, Jeremy Clark, Sarah Meiklejohn, Peter Y. A. Ryan, Dan S. Wallach, Michael Brenner, and Kurt Rohloff (Eds.). Springer, 19–26. https://doi.org/10.1007/978-3-662-53357-4_2

[18] Lars Brünjes, Aggelos Kiayias, Elias Koutsoupias, and Aikaterini-Panagiota Stouka. 2018. Reward Sharing Schemes for Stake Pools. *CoRR* abs/1807.11218 (2018). arXiv:1807.11218 http://arxiv.org/abs/1807.11218

[19] Ran Canetti. 2000. Security and Composition of Multiparty Cryptographic Protocols. *Journal of Cryptology* 13, 1 (01 Jan 2000), 143–202. https://doi.org/10.1007/s001459910006

[20] Ran Canetti. 2000. Universally Composable Security: A New Paradigm for Cryptographic Protocols. Cryptology ePrint Archive, Report 2000/067. https://eprint.iacr.org/2000/067.

[21] R. Canetti. 2001. Universally Composable Security: A New Paradigm for Cryptographic Protocols. In *Proceedings of the 42Nd IEEE Symposium on Foundations of Computer Science (FOCS '01)*. IEEE Computer Society, Washington, DC, USA, 136–145. http://dl.acm.org/citation.cfm?id=874063.875553

[22] Miles Carlsten, Harry Kalodner, S. Matthew Weinberg, and Arvind Narayanan. 2016. On the Instability of Bitcoin Without the Block Reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. ACM, New York, NY, USA, 154–167. https://doi.org/10.1145/2976749.2978408

[23] Cynthia Dwork and Moni Naor. 1993. Pricing via Processing or Combatting Junk Mail. In *Advances in Cryptology — CRYPTO' 92*, Ernest F. Brickell (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 139–147.

[24] Ittay Eyal and Emin Gün Sirer. 2014. Majority Is Not Enough: Bitcoin Mining Is Vulnerable. In *Financial Cryptography and Data Security*, Nicolas Christin and Reihaneh Safavi-Naini (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 436–454.

[25] Amos Fiat, Anna Karlin, Elias Koutsoupias, and Christos Papadimitriou. 2019. Energy Equilibria in Proof-of-Work Mining. In *Proceedings of the 2019 ACM Conference on Economics and Computation* (Phoenix, AZ, USA) *(EC '19)*. ACM, New York, NY, USA, 489–502. https://doi.org/10.1145/3328526.3329630

[26] Juan Garay, Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. 2013. Rational Protocol Design: Cryptography Against Incentive-Driven Adversaries. In *Proceedings of the 2013 IEEE 54th Annual Symposium on Foundations of Computer Science (FOCS '13)*. IEEE Computer Society, Washington, DC, USA, 648–657. https://doi.org/10.1109/FOCS.2013.75

[27] Juan Garay, Aggelos Kiayias, and Nikos Leonardos. 2015. The Bitcoin Backbone Protocol: Analysis and Applications. In *Advances in Cryptology - EUROCRYPT 2015*, Elisabeth Oswald and Marc Fischlin (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 281–310.

[28] A. Gervais, G. O. Karame, V. Capkun, and S. Capkun. 2014. Is Bitcoin a Decentralized Currency? *IEEE Security Privacy* 12, 3 (May 2014), 54–60. https://doi.org/10.1109/MSP.2014.49

[29] Arthur Gervais, Ghassan O. Karame, Karl Wüst, Vasileios Glykantzis, Hubert Ritzdorf, and Srdjan Capkun. 2016. On the Security and Performance of Proof of Work Blockchains. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. ACM, New York, NY, USA, 3–16. https://doi.org/10.1145/2976749.2978341

[30] Adam Groce, Jonathan Katz, Aishwarya Thiruvengadam, and Vassilis Zikas. 2012. Byzantine Agreement with a Rational Adversary. In *Automata, Languages, and Programming*, Artur Czumaj, Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 561–572.

[31] Cyril Grunspan and Ricardo Pérez-Marco. 2018. On profitability of selfish mining. *CoRR* abs/1805.08281 (2018). arXiv:1805.08281 http://arxiv.org/abs/1805.08281

[32] Jr. Harvey S. James. 2014. *Incentive compatibility.* Encyclopedia Britannica, inc. Encyclopedia Britannica https://www.britannica.com/topic/incentive-compatibility.

[33] Charlie Hou, Mingxun Zhou, Yansquir Ji, Phil Daian, Florian Tramer, Giulia Fanti, and Ari Juels. 2019. SquirRL: Automating Attack Discovery on Blockchain Incentive Mechanisms with Deep Reinforcement Learning. arXiv:1912.01798 [cs.CR]

[34] Edward W Felten Joshua A Kroll, Ian C Davey. 2013. The economics of Bitcoin mining, or Bitcoin in the presence of adversaries. In *Proceedings of WEIS*.

[35] Ari Juels and John G. Brainard. 1999. Client Puzzles: A Cryptographic Countermeasure Against Connection Depletion Attacks. In *NDSS*. The Internet Society.

[36] Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. 2013. Universally Composable Synchronous Computation. In *Theory of Cryptography*, Amit Sahai (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 477–498.

[37] Yoav Shoham Kevin Leyton-Brown. 2008. *Essentials of Game Theory: A Concise Multidisciplinary Introduction (Synthesis Lectures on Artificial Intelligence and Machine Learning).* Morgan & Claypool Publishers.

[38] Aggelos Kiayias, Elias Koutsoupias, Maria Kyropoulou, and Yiannis Tselekounis. 2016. Blockchain Mining Games. In *Proceedings of the 2016 ACM Conference on Economics and Computation* (Maastricht, The Netherlands) *(EC '16)*. ACM, New York, NY, USA, 365–382. https://doi.org/10.1145/2940716.2940773

[39] Aggelos Kiayias and Aikaterini-Panagiota Stouka. 2019. Coalition-Safe Equilibria with Virtual Payoffs. arXiv:2001.00047 [cs.GT]

[40] Abhiram Kothapalli, Andrew Miller, and Nikita Borisov. 2017. SmartCast: An Incentive Compatible Consensus Protocol Using Smart Contracts. In *Financial Cryptography and Data Security*, Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y.A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, and Markus Jakobsson (Eds.). Springer International Publishing, Cham, 536–552.

[41] Leslie Lamport, Robert E. Shostak, and Marshall C. Pease. 1982. The Byzantine Generals Problem. *ACM Trans. Program. Lang. Syst.* 4, 3 (1982), 382–401. https://doi.org/10.1145/357172.357176

[42] Nikos Leonardos, Stefanos Leonardos, and Georgios Piliouras. 2019. Oceanic Games: Centralization Risks and Incentives in Blockchain Mining. *CoRR* abs/1904.02368 (2019). arXiv:1904.02368 http://arxiv.org/abs/1904.02368

[43] Stefanos Leonardos, Daniel Reijsbergen, and Georgios Piliouras. 2019. PREStO: A Systematic Framework for Blockchain Consensus Protocols. arXiv:1906.06540 [cs.CR]

[44] Yoad Lewenberg, Yonatan Sompolinsky, and Aviv Zohar. 2015. Inclusive Block Chain Protocols. In *Financial Cryptography and Data Security*, Rainer Böhme and Tatsuaki Okamoto (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 528–547.

[45] Kevin Liao and Jonathan Katz. 2017. Incentivizing Blockchain Forks via Whale Transactions. In *Financial Cryptography and Data Security*, Michael Brenner, Kurt Rohloff, Joseph Bonneau, Andrew Miller, Peter Y.A. Ryan, Vanessa Teague, Andrea Bracciali, Massimiliano Sala, Federico Pintore, and Markus Jakobsson (Eds.). Springer International Publishing, Cham, 264–279.

[46] J. Liu, W. Li, G. O. Karame, and N. Asokan. 2018. Toward Fairness of Cryptocurrency Payments. *IEEE Security Privacy* 16, 3 (May 2018), 81–89. https://doi.org/10.1109/MSP.2018.2701163

[47] Ziyao Liu, Nguyen Cong Luong, Wenbo Wang, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. 2019. A Survey on Applications of Game Theory in Blockchain. arXiv:1902.10865 [cs.GT]

[48] Loi Luu, Jason Teutsch, Raghav Kulkarni, and Prateek Saxena. 2015. Demystifying Incentives in the Consensus Computer. In *Proceedings of the 22Nd ACM SIGSAC Conference on Computer and Communications Security* (Denver, Colorado, USA) *(CCS '15)*. ACM, New York, NY, USA, 706–719. https://doi.org/10.1145/2810103.2813659

[49] Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. 2013. A Fistful of Bitcoins: Characterizing Payments Among Men with No Names. In *Proceedings of the*

2013 Conference on Internet Measurement Conference (Barcelona, Spain) *(IMC '13)*. ACM, New York, NY, USA, 127–140. https://doi.org/10.1145/2504730.2504747

[50] Satoshi Nakamoto. 2008. *Bitcoin: A Peer-to-Peer Electronic Cash System.* http://bitcoin.org/bitcoin.pdf.

[51] K. Nayak, S. Kumar, A. Miller, and E. Shi. 2016. Stubborn Mining: Generalizing Selfish Mining and Combining with an Eclipse Attack. In *2016 IEEE European Symposium on Security and Privacy (EuroS P)*. 305–320. https://doi.org/10.1109/EuroSP.2016.32

[52] Rafael Pass, Lior Seeman, and Abhi Shelat. 2017. Analysis of the Blockchain Protocol in Asynchronous Networks. In *Advances in Cryptology – EUROCRYPT 2017*, Jean-Sébastien Coron and Jesper Buus Nielsen (Eds.). Springer International Publishing, Cham, 643–673.

[53] Rafael Pass and Elaine Shi. 2017. FruitChains: A Fair Blockchain. In *Proceedings of the ACM Symposium on Principles of Distributed Computing* (Washington, DC, USA) *(PODC '17)*. ACM, New York, NY, USA, 315–324. https://doi.org/10.1145/3087801.3087809

[54] R. L. Rivest, A. Shamir, and D. A. Wagner. 1996. *Time-lock Puzzles and Timed-release Crypto.* Technical Report. Cambridge, MA, USA.

[55] Dorit Ron and Adi Shamir. 2013. Quantitative Analysis of the Full Bitcoin Transaction Graph. In *Financial Cryptography and Data Security*, Ahmad-Reza Sadeghi (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 6–24.

[56] Tim Roughgarden. 2020. Transaction Fee Mechanism Design for the Ethereum Blockchain: An Economic Analysis of EIP-1559. arXiv:2012.00854 [cs.GT]

[57] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. 2017. Optimal Selfish Mining Strategies in Bitcoin. In *Financial Cryptography and Data Security*, Jens Grossklags and Bart Preneel (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 515–532.

[58] Fred B. Schneider. 1990. Implementing Fault-Tolerant Services Using the State Machine Approach: A Tutorial. *ACM Comput. Surv.* 22, 4 (1990), 299–319. https://doi.org/10.1145/98163.98167

[59] Itay Tsabary and Ittay Eyal. 2018. The Gap Game. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (Toronto, Canada) *(CCS '18)*. ACM, New York, NY, USA, 713–728. https://doi.org/10.1145/3243734.3243737

[60] Itay Tsabary and Ittay Eyal. 2018. The Gap Game. *CoRR* abs/1805.05288 (2018). arXiv:1805.05288 http://arxiv.org/abs/1805.05288