PRAGTHOS: PRACTICAL GAME THEORETICALLY SECURE PROOF-OF-WORK BLOCKCHAIN

A PREPRINT

Varul Srivastava
Machine Learning Lab
IIIT Hyderabad
varul.srivastava@research.iiit.ac.in

Dr. Sujit Gujar
Machine Learning Lab
IIIT Hyderabad
sujit.gujar@iiit.ac.in

February 13, 2023

ABSTRACT

Security analysis of blockchain technology is an active domain of research. There has been both cryptographic and game-theoretic security analysis of Proof-of-Work (PoW) blockchains. Prominent work includes the cryptographic security analysis under the Universal Composable framework and Game-theoretic security analysis using Rational Protocol Design. These security analysis models rely on stricter assumptions that might not hold.

In this paper, we analyze the security of PoW blockchain protocols. We first show how assumptions made by previous models need not be valid in reality which an attacker can exploit to launch attacks that these models fail to capture. These include (1) Difficulty Alternating Attack, under which forking is possible for an adversary with $< \frac{1}{2}$ mining power, (2) Quick-Fork Attack, (3) a general bound on selfish mining attack and, (4) transaction withholding attack. Following this, we argue why previous models for security analysis fail to capture these attacks and propose a more practical framework for security analysis – pRPD. We then propose a framework to build PoW blockchains **PRAGTHOS**, which is secure from the attacks mentioned above. Finally, we argue that PoW blockchains complying with the **PRAGTHOS** framework are secure against a computationally bounded adversary under certain conditions on the reward scheme.

1 Introduction

Blockchain technology was introduced by Satoshi Nakamoto Nakamoto [2009] via Bitcoin – an alternative to centralized financial institutions. Blockchain is an immutable *distributed* ledger that achieves distributed trust. It stores the data in blocks connected through certain cryptographic links and ensures consensus on a single state (ordering of blocks) across all participants. Vitalik Buterin et al. [2013] introduced the Ethereum blockchain, which supports Turing complete smart-contract functionality Szabo [1997]. Since then, blockchain technology has found immense applications in multiple domains, such as financial, supply-chains, Smart-city and IoT systems, Decentralized Governance, Voting and Auction systems, etc. Blockchain technology *distributes* trust by maintaining *consensus* over the state of the system.

The FLP impossibility theorem (Fischer et al. [1983]) states that a deterministic algorithm cannot achieve consensus in an asynchronous system even if a single node is faulty. Blockchains are *decentralized* systems that use *incentive-engineering* to overcome these impossibilities. It uses *Proof-of-Work* (PoW) consensus algorithm to ensure consistency and overcome such impossibilities otherwise existent in distributed systems. In PoW, each participating node (miner) is supposed to solve a certain cryptographic puzzle through queries to write to the blockchain. The query is successful if the block's hash value is lesser than some decided target. Blockchain protocols overcome impossibilities because (i) they are incentive-based protocol – offers rewards to the miner who writes the block (ii) they are non-deterministic.

As blockchain protocols gained popularity, researchers have done extensive security analysis of PoW-based blockchains Garay et al. [2015, 2017], Eyal and Sirer [2014]. Badertscher et al. Badertscher et al. [2017] use the *Universal Composable (UC) Framework* (introduced by Canetti Canetti [2001]) to propose a universal composable treatment of Bitcoin. However, in the security analysis of blockchains, we should consider not only *cryptographic* but also *game-theoretic* security. Garay et al. Garay et al. [2013] proposed *Rational Protocol Design* (RPD) for game-theoretic security analysis of any distributed cryptographic protocol. Badertscher et al. Badertscher et al. [2018, 2021] modified the RPD model for game-theoretic security analysis of PoW blockchains. There has been extensive research in the domain of game-theoretic security of PoW blockchain Judmayer et al. [2020], Liao and Katz [2017a,b], Han et al. [2021], Tsabary and Eyal [2018], Siddiqui et al. [2020], Chen et al. [2021], Karakostas et al. [2022], Azouvi and Hicks [2020]. In particular, Badertscher et al. Badertscher et al. [2018] proves that Bitcoin satisfies strong notions of security such as*strong-attack-payoff-security*[1] proposed in Garay et al. [2013] under honest majority assumption. However, this work demonstrates how an attacker can attack Bitcoin or many PoW blockchains even if it has $< 50\%$ computing power using DIFFICULTY ALTERING attack.

The primary reason for the existing frameworks could conclude security is that they assume one or more of the following: (i) constant conversion rate of crypto-currency to fiat currency, (ii) constant block reward, or (iii) constant difficulty. In reality, these are dynamic parameters. Additionally, these works categorize miners into *honest party* (HP) – who follow the protocol honestly or *advesarial party* (AP)– who launch attacks on the protocol. Some miners typically follow the protocol but may deviate conditionally if a strategy offers a higher utility without disrupting the blockchain. We call such miners *Rational Party* (RP). In this work, we show that multiple attacks are possible in the presence of RP on the PoW blockchain, which RPD-based models fail to capture. Hence, our goal is to build a practical model that aligns with the real-world environment for game-theoretic security analysis of a PoW blockchain exists.

Our Contributions

Our contributions in this work are threefold. (i) We identify specific attacks on existing PoW blockchain protocols and explain why existing security models fail to capture them. (ii) We propose a new model – PRACTICAL RATIONAL PROTOCOL DESIGN (PRPD) to overcome the limitations of existing models for PoW blockchain. (iii) We propose solutions to overcome the attacks identified, propose a novel framework **PRAGTHOS** and perform security analysis of PoW blockchains complying with the framework.

Attacks: For PoW blockchains, we show that there exist previously undiscovered attacks. Additionally, on known attacks, we prove that the fraction of adversarial computing power required to launch the existing attack is much smaller than the existing bounds.

- We prove that there exist DIFFICULTY ALTERING in which AP with $< \frac{1}{2}$ can launch a successful fork without the help of RP. The previous estimate of this bound was $\frac{1}{2}$ (51% attack). As a corollary, such an attack is possible in Bitcoin even when AP controls 45% of the mining power (Theorem 3.1).
- We show that there exists a security attack QUICK FORK where AP leverages incentive-driven deviations from RP (Theorem 3.2).
- We also show that the bound for selfish-mining proposed in Eyal and Sirer [2014] is a special case of a more general bound. Using our proposed model PRPD, we can capture this more general bound when the selfish-mining attack is combined with whale-transactions Liao and Katz [2017b], which we call – SELFISH MINING WITH BRIBING.
- We show that under transaction-fee only model (TFOM)[2], there exists a deviation from *Gossip* protocol (Lemma 3.4) which we call a transaction-withholding attack. This strategic deviation can either lower the throughput of the PoW blockchain or centralize the protocol.

PRPD: We discuss why previous works fail to capture these security attacks and make corresponding modifications to introduce a more practical model for security analysis of PoW blockchain protocol – PRPD. In this model, we account for multiple mining rounds while also capturing the system's dynamics – variable block reward and variable difficulty. We also capture the external responses to the system through externalities (change in conversion rate to fiat currency), which depend on the strategies followed by the miners. Further, we include three types of players, altruistic (honest), conditionally-deviating (rational), and deviating (adversarial) in our analysis.

Detering Attacks & PRAGTHOS: To rectify the security attacks above, we propose certain amendments that a PoW blockchain protocol should adopt. Combining these modifications, we propose a novel framework to build a PoW blockchain, namely **PRAGTHOS**.

- We determine appropriate hyper-parameter values to ensure DIFFICULTY ALTERING attack is not possible (Corr. 5.1).
- We propose `PC-MOD` as a deterrence to QUICK FORK attack. We show that the existence of such a protocol (to be run in case QUICK FORK attack is launched) is sufficient to prevent the attack.

[1] see Definition 4.2 in this paper

[2] TFOM formally explained in Appendix D

- We also show a pessimistic result that there does not exist any PoW blockchain protocol that is attack-payoff secure against Selfish mining (Theorem 5.2).
- We propose $\Pi_{tx-inclusion}$ protocol to ensure that deviation from *Gossip* protocol is disincentivized.
- Based on these modifications, we introduce **PRAGTHOS** and perform security analysis.
  - Under inflationary block-reward scheme we show that **PRAGTHOS** is *strongly attack-payoff secure* (Theorem 6.1).
  - We show that under the deflationary block-reward scheme: (1) no PoW blockchain protocol is *strongly attack-payoff secure* against a general adversary (Theorem 6.3). (2) **PRAGTHOS** is secure against an adversary with an attack strategy bounded in the number of rounds (Theorem 6.4).

With PRPD and **PRAGTHOS**, we have significant results : (i) Bitcoin is not secure even with the honest majority, (ii) Crypto-currencies with deflationary reward schemes are not strongly attack-payoff secure. Thus, we believe this work lays the foundations for building future PoW blockchain protocols.

Related Work

The analysis of cryptographic and game-theoretic security of PoW blockchain protocols have been extensively done in recent literature, and multiple attacks have been discovered and discussed. In this section, we contrast our work with the existing attacks and the analysis frameworks.

General Security. Tsabary and Eyal discuss in Tsabary and Eyal [2018] that if the subsidy (cumulation of transaction fee and block-reward) is small enough and the cost of mining is high enough, then the gap between intervals where miners mine blocks increases. Further, this gap varies based on the relative size of mining pools, and it is incentivized to form coalitions to increase the reward. This means that sparse transaction distribution and lesser block rewards lead to the centralization of the system. Wei et al. Wei et al. [2018] study the effect of network delays in Blockchain networks. The authors propose a more general framework to model network delays, with large possible delays happening probabilistically. However, in their analysis, the adversary is restricted to performing network delays. Gazi et al. Gazi et al. [2022] proposed an efficient method to compute bounds on settlement time for PoW blockchain given computation power and network delays. Yuan et al. Yuan et al. [2020] discuss the protocol with $(1 + \mu)$ miners, out of which μ are corrupted. Further, the adversary can delay the network by at most $\frac{1}{np\Delta}$ with probability 1. In this case, the authors discuss conditions under which chain growth, chain quality, and common prefix properties hold[3]. Momeni et al. Momeni et al. [2022] propose an identity-based encryption mechanism to prevent front-running. However, this solution is not efficiently applicable in public PoW blockchains because the protocol is based on the committees' identities that perform consensus. There exist works like Badertscher et al. [2017], Graf et al. [2021] which perform universal composable treatment of Blockchains. The general framework of cryptographic security analysis discussed in Graf et al. [2021] is applicable to general distributed ledgers, whereas our work analyses game-theoretic security as well. Ke et al.(Ke et al. [2020]) model contingency plans for severe attacks on blockchains. They propose a framework that inputs the attack on the blockchain and gives the contingency plan, detection plan, and level of damage as output. Their work proposes frameworks for contingency plans and assessing the level of damage for the given attack, unlike our work which focuses on modeling the system to capture all possible attacks and proposing a framework to resolve those attacks.

Rational Security. Han et al. Han et al. [2021] points out that the honest majority assumption is not true, especially for some miners who may be adversaries who can be lured to work on a fork through incentive engineering. The authors show that there exist incentive structures if we consider the existence of multiple blockchains in the system, such that miners can be incentivized to launch 51% attacks on the blockchain with less computing power. Their work differs from ours because they consider systems with multiple blockchains. We discuss multiple attacks, which are possible even in a blockchain system where the majority of miners are honest. Kevin and Katz (Liao and Katz [2017b],Liao and Katz [2017a]) performed one of the earliest works which discuss the idea of (non-adversarial) RP who can deviate from the longest chain. Their attack is through *Whale Transactions* – transactions with abnormally high transaction fees. Judmayer et al. (Judmayer et al. [2020], Han et al. [2021]), consider an adversary who bribes RP through external guaranteed incentives through smart contracts for joining the attack – the authors call it *Algorithmic Incentive Manipulations*. Our model uses RP to capture a larger pool of miners as they still may deviate even in the absence of bribes. Thus, our model is more general. Another line of attacks is when Block Reward reduces to become negligibly small and miner incentive is driven by Transaction Fees only. Most of the existing works Karakostas et al. [2022], Siddiqui et al. [2020], Möser and Böhme [2015], Li et al. [2018], Houy [2014], Roughgarden [2021] either study the incentive-based deviations of transaction proposers, or makes optimistic assumptions on the distributions of Transactions. Badertscher et al. Badertscher et al. [2018] and Carlsten et al. Carlsten et al. [2016] show that under

[3] there 3 properties are required for blockchain security

an unfavorable (but equally possible) distribution of incoming transactions, the Protocol is not secure. We discuss a possible deviation that compromises the protocol under such transaction distribution and how to tackle such deviation.

Security Models. Badertscher et al. Badertscher et al. [2018, 2021] performed RPD on Bitcoin and analysis on double-spending in case of 51% attack. These works fail to capture attacks that enable Forking and Double-spending for an adversary with $\beta_{adv} < \frac{1}{2}$ without leveraging deviations from other miners. The utility model in Badertscher et al. [2018] fails to capture variable block reward, variable conversion rate, and the variable cost of mining. The authors generalized the utility model in RPD to capture variable block reward and cost of mining in Badertscher et al. [2021]. The authors argue that due to decreasing block-reward, *attack-payoff-security* should be considered only for *finite-horizon*. As the conversion rate (from cryptocurrency to fiat) is also dynamic (dependent on participating players' strategies and supply of the cryptocurrency). Note that this conversion rate can increase[4], rendering the finite horizon argument made in Badertscher et al. [2021] impractical. On the other hand, our model captures both discounting and deflationary block rewards. Thus, our results capture a more realistic behavior of the system.

2 Preliminaries

In this section, we discuss the background, notations, and formalism necessary for the discussions that follow in this paper.

2.1 Universal Composability

First, we brief the set of protocols that are considered when defining security for PoW blockchains. Canetti Canetti [2001] proposes the framework of *Universal Composability* (UC), and later Badertscher et al. [2017] proposes a UC treatment of PoW blockchain. Similar to the previous works, our model assumes all participating parties are *Probabilistic Polynomial Time* (PPT) *Interactive Turing Machines* (ITMs).

2.1.1 $\mathcal{G}_{ledger}$ Ideal Functionality

Ideal Functionality is the functionality that the proposed protocol aims to achieve. A PoW blockchain aims to achieve $\mathcal{G}_{ledger}$ functionality as described in Badertscher et al. [2017]. This functionality stores a ledger that maintains the system's state, with each state transition performed through transactions submitted by participating parties. Different parties might have different points of view for the state head; therefore, the functionality stores a pointer for the state head according to each party. The `VALID-TX` predicate checks the validity of the transactions and appends them to the ledger if valid (makes corresponding state transactions). The predicate enforces *ledger growth*, *chain quality*, and *transaction liveness*.

2.1.2 $\mathcal{G}_{weak-ledger}$ Relaxed Functionality

The concept of relaxing the ideal functionality exists because there are multiple attacks and other behaviors in the real-world which need to be simulated in the ideal world. The relaxed, ideal world functionality of $\mathcal{G}_{ledger}$ is defined as, $\mathcal{G}_{weak-ledger}$ in Badertscher et al. [2018]. The relaxed functionality stores the states as a tree instead of a linked list, which allows forks of arbitrary lengths to exist, and we choose the longest chain out of these forks, as in all bitcoin-like PoW blockchains.

Further, since in real world it is indistinguishable if the block was mined by an adversarial, rational or honest miner, the functionality relaxes checks on the chain quality and chain growth properties, which are verified in `StateExtend` policy. In $\mathcal{G}_{weak-ledger}$ the `StateExtend` policy is relaxed to `WeakStateExtend`. The relaxed functionality accepts all transactions in the state buffer without checking their validity against the state-tree because these transactions might be valid in one of the multiple chains of the state-tree. In addition, the ability to create forks can be invoked by the `Fork` command, which extends the chain from an indicated block, instead of the traditional `Next Block` command, which extends the chain from the header for the calling party. (For more details on $\mathcal{G}_{ledger}$ and $\mathcal{G}_{weak-ledger}$, please refer to Badertscher et al. [2017] and Badertscher et al. [2018] respectively.)

2.2 PoW Blockchain Protocol

A PoW blockchain consists of a chain and a hash-pointer-based linked list. Further, to add a block, we have to include the hash of the parent block to which it points and solve a PoW puzzle (by choosing a nonce) such that the block's hash

[4] as is visible for Bitcoin and Ethereum's historical prices

is less than a target. A lower target indicates the greater difficulty of the puzzle. We represent a PoW blockchain that UC-realizes (refer Def. 20, Canetti [2001]) the relaxed ideal functionality as Π.

Let $C_i^{\lfloor k}$ be the subset of all but the last k blocks in chain C_i. We consider additional and practical relaxation of G_{ledger} ideal functionality, namely variable difficulty. The probability of mining a block on different chains can vary based on the chain's difficulty. The difficulty of mining a block on any C_i is publicly known, as it can be calculated using the publically available ledger. The sub-protocol $isValidStruct(C)$, defined in Section 5.1 of Badertscher et al. [2017], takes the difficulty parameter corresponding to the chain given as input instead of the system parameter. The system rewards miners with a block reward for maintaining a ledger by solving PoW puzzles.

2.2.1 Ledger

The ledger contains blocks that a miner mines. The block $b = (h_{parent}, nonce, txMD, \{0,1\}^*)$ contains (i) hash pointer to the parent block h_{parent}, (ii) a random nonce $nonce$ and, (iii) the meta-data for the set of transactions to be included. This metadata depends upon the blockchain implementation details. E.g., it includes the root of the Merkle tree containing a set of transactions, which in turn also include the coinbase transaction[5] in Bitcoin Nakamoto [2009]. Further, the contract can include other information in the form of arbitrary binary strings to enable additional features in the ledger.

2.2.2 Timesteps

Since there is no global clock in the system (which runs in a partially-synchronous setting Dwork et al. [1984]), we cannot divide the measurement of events in terms of time. We use the notion of *rounds*, which depends on the number of hashes computed by the system.

Definition 2.1 (Round)**.** A *round* is a duration in which any miner makes q queries[6].

In addition to *rounds*, we define an *epoch* to capture the event of the change in difficulty of mining and *phase* to denote the change in block-reward (e.g. in Bitcoin, reward halves every $210,000$ blocks).

Definition 2.2 (Epoch)**.** An *epoch* is a duration in which the system mines λ blocks at constant difficulty. The difficulty is scalled by τ at the end of an epoch for $\tau \in [\tau_{min}, \tau_{max}]$.

Definition 2.3 (Phase)**.** A *phase* is a duration in which the system mines Λ blocks. At the end of a phase j, block reward gets updated by the update rule $r_{block}^{new} \leftarrow \varrho(r_{block}^{old}, j)$[7] for some $\varrho : \mathbb{R} \times \mathbb{N} \rightarrow \mathbb{R}$

The protocol execution comprises different parties participating in each round, whose roles are explained below.

2.2.3 Parties

PoW-based blockchains have three types of parties – Altruistic (Honest), Adversarial and Rational. Each party consists of a set of miners who exhibit the same behavior. In our formulation, to model market responses, we introduce a dummy, passive party, an *External observer*.

External Observer(EO) acts as an observer of public chains of the blockchain. We assume a *tolerance factor* ρ to account for network latency and accidental forks. That is, EO considers a forked chain C_f as a forking (security) attack only if there existed another chain C_h which previously had a lead of ρ blocks and C_f overtakes C_h eventually. In this work, we model the drastic conversion rate changes that can happen within one round and not the slow ones which we observe. In round t, EO sends a signal $\theta(t)$ to the parties. If $\theta(t) = 1$ implies that there is no major disruption in the conversion rate.

$$\theta(t) = \begin{cases} 1 & \text{if all follow protocol honestly} \\ e_{fairness} & \text{if fairness attacks} \\ e_{security} & \text{if security attacks} \end{cases} \tag{1}$$

1. **Honest Party(HP):** These miners control β_{hon} fraction of total mining power. HP participate in the system by following the PoW blockchain protocol honestly if participation is profitable; otherwise, they do not participate.

[5] Bitcoin's mechanism to pay miners a block reward

[6] in PoW blockchain these are hash-queries

[7] for Bitcoin $\varrho(r_{block}^{old}, j) = \frac{r_{block}^{old}}{2}$

2 **Adversarial Party(AP):** These miners control β_{adv} fraction of the total mining power. Adversarial miners launch attacks and deviate from the honest protocol. They received payoff through the crypto-currency as well as by short-selling[8]. Thus the change in $\theta(t)$ is inconsequential for AP.

3 **Rational Party(RP):** These miners control β_{rat} fraction of total mining power and follow the protocol – Π unless there exists a deviation with higher utility. However, this party would deviate only if the deviation cannot be a security attack observed by EO, i.e., it is guaranteed that after deviation $\theta(t) \neq e_{security}$.

Note that we consider all miners to be computationally bounded and have a fixed computing power. However, if a single player increases its mining power, we consider it multiple miners (without loss of generality); each making q queries in a round.

Next, we explain the role of EO in modeling the market response, conversion rates, etc.

2.2.4 Modelling Externalities

In PRPD, externality plays an important role while modeling the behavior of rational players. Badertscher et al. Badertscher et al. [2018] assign a very high (exponential in poly-log of security factor) negative payoff to the Protocol Descriptor to model the effect on the protocol (or the value of the cryptocurrency). In our case, we have defined the term $\theta(t)$, which changes the crypto-currency value based on the strategies miners follow, as observed by EO – the value reduces to < 1 if there is deviation. If the payoff in cryptocurrency is $\mathcal{R}$, then the actual payoff is $\theta(t)CR\mathcal{R}$. We model the market response to different strategies through $\theta(t)$ and the supply-demand fluctuation is handled by CR. We divide the attacks on the blockchain system into two categories.

1 **Fairness-Attacks** compromise on fairness, i.e., some miner gaining more rewards than its fair share or some user needing to wait for a transaction to be accepted unreasonably high. In these attacks, the security remains intact. We assume if such an attack is observed by EO, $\theta(t) = e_{fairness} < 1$.

2 **Security-Attacks** is when security of the protocol is compromised. In this case, the externality parameter $\theta(t) = e_{security}$.[9]

For example, security threat can be modeled as $e_{security} = negl(\kappa)$. In that case, $e_{security} = \frac{1}{poly(\kappa)}$.

2.3 Existing Attacks on PoW Blockchain

There have been several attacks proposed Eyal and Sirer [2014], Liao and Katz [2017a,b], Eskandari et al. [2019], Breidenbach et al. [2018], Kalodner et al. [2015] for PoW based blockchain. We enlist the previously discovered attacks which are relevant to our work.

2.3.1 Selfish Mining

Selfish mining is an attack proposed by Eyal and Sirer Eyal and Sirer [2014]. In this attack, the adversary gains a higher fraction of the total block reward for any continuous set of blocks mined than the fraction of total mining power held by the adversary. Therefore, this is an attack on the system's fairness, not a security attack. However, the authors of Eyal and Sirer [2014] show that after certain modifications, only an adversary with $> \frac{1}{4}$ mining power can successfully launch a selfish mining attack.

Claim 2.1 (Eyal and Sirer [2014], Observation 1)**.** *For a given γ, an adversarial pool of size β_{adv} obtains a revenue larger than the relative size for β_{adv} in the following range:*

$$\frac{1-\gamma}{3-\gamma} < \beta_{adv} < \frac{1}{2} \tag{2}$$

Where γ is the fraction of non-adversarial parties that mine on top of the adversarial block in case of competition among two blocks for the longest chain.

Carlsten et al. Carlsten et al. [2016] discuss the possibility of Selfish Mining being profitable for an adversary with arbitrarily low mining power. Their result holds only in Transaction Fee Only Model (TFOM– when block rewards are negligible) and relies on the non-uniformity of rewards. Our work proposes that in the presence of RP, an adversary with arbitrarily low mining power might be incentivized to launch a Selfish mining attack and is possible when there are block significant rewards.

[8] short-selling is typical terminology in stock-trading; one sells the stocks (in this case, cryptocurrency) not owned by it and repurchases at a lower price in the future.

[9] $e_{fairness} >> e_{security}$ because security attacks are a more serious threat.

2.3.2 Forking Attack

A forking attack is a security attack on the PoW blockchain. We say a chain C_A overtaking chain C_H at some time is a forking attack if (1) at some time t_1, $length(C_H - C_A) \geq k$ (where k is a parameter set by the PoW blockchain[10]. (2) C_A overtakes C_H at some time $t_2 > t_1$. Such attacks can lead to double-spending and are, therefore, a serious security threat to the blockchain.

2.3.3 Timewarp Attack

Several adversarial manipulations exist such as the Timewarp-attack Friedenbach [2018] that uses incorrect time-stamping to reduce mining difficulty. However, Timewarp-attack is feasible only if adversary holds $> 51\%$ of the mining power. PoW blockchains with difficulty recalculation each round (like the Verge Shirah [2008]) suffer from security threat. However, PoW blockchains like the Bitcoin are secure against timewarp attack under honest majority. In contrast, the DIFFICULTY ALTERING attack (Section 3.1) are possible in Bitcoin and similar blockchains even if majority is honest.

2.3.4 Front-running attacks

We assumpe a special class of adversarial PPT ITMs $\mathcal{A}^*_{fr}$ –*front-running* adversaries, originally defined in Badertscher et al. [2018]

Definition 2.4 (Front-Running, Def. 2, Badertscher et al. [2018])**.** An adversary $A \in \mathcal{A}^*_{fr}$ (is a front-running adversary) if it satisfies the following conditions:

1. Upon receiving a broadcast message by a party, it can maximally delay it by one round.
2. Any broadcast message by the adversary is propagated through the network immediately.

Under such an adversary, there exist attacks discussed in Eskandari et al. [2019], Breidenbach et al. [2018], Kalodner et al. [2015] such as (1) *displacement* attack and (2) *insertion* attacks. For our purposes, however, the Definition 2.4 is of interest.

2.4 Rational Protocol Design

Rational Protocol Design Garay et al. [2013] (RPD) is the basis on which multiple security analysis models for PoW blockchain protocols is based, including Badertscher et al. [2018, 2021]. Our work motivates from the RPD and modifies it in context of PoW Blockchain protocols to a more practical model of Game Theoretic security analysis.

RPD models security as a game between two players, (1) the Protocol Descriptor (PD) which proposes the protocol that realizes the relaxed functionality and, (2) the Adversary, which chooses attack strategy once the PD has chosen a protocol. The Game $\mathcal{G}_{\mathcal{M}}$ is modelled as a two-player game with complete information and finite horizon (of two steps) – a Stackelberg Game. RPD models deviating (adversaries) and non-deviating (altruistic) players.

Badertscher et al. Badertscher et al. [2018] modified RPD protocol curating to PoW blockchains. This protocol captured deviations by the protocol across rounds. However, this model didnot capture dynamics of the system such as variable mining difficulty which can be strategically manipulated by the adversary (as we see in Section 3.1), variable block-reward among other things. Further, the game $\mathcal{G}_{\mathcal{M}}$ models deviating (adversarial) and non-deviating (altruistic) players. They do not capture *conditionally-deviating* (rational) players. We aim at introducing a Practical Rational Protocol Design model (pRPD Section 4), which can model such setting.

3 Attacks on PoW blockchains

Previous models of security analysis such as Garay et al. [2013], Badertscher et al. [2018], Karakostas et al. [2022] are good contributions. Still, their analysis is limited to a fixed difficulty, finite horizons, constant block reward, constant conversion rate, and a single mining round. In practical scenarios, these factors are dynamic, and an adversary can leverage them to launch attacks. This section discusses the attacks we have discovered on PoW-based blockchains that previous works fail to capture. We also discuss why previous works fail to capture these attacks.

We categorize these attacks into three categories, (i) *Byzantine* adversary attacks – the worst form of attack where a byzantine adversary with $\beta_a < \frac{1}{2}$ can single-handedly launch the attack. (ii) *Rational-Byzantine* attacks – where a

[10] for Bitcoin $k = 6$

byzantine adversary relies on rational agents to successfully launch an attack. (iii) *Rational* attacks – deviations of Rational Party from the original protocol Π. In these attacks, no byzantine adversary is involved.

3.1 Byzantine adversary attacks

We discovered that a byzantine adversary could launch a forking attack and double spend. Previously it was considered that if $\beta_a < \frac{1}{2}$, then the protocol is considered secure against such double spending attacks with a very high probability for reasonable block-confirmation time. However, we show that even for a very liberal block confirmation time, there exists an adversarial attack that can cause forks with very high probability even for $\beta_a < \frac{1}{2}$.

3.1.1 Difficulty altering attack

The DIFFICULTY ALTERING takes place in two consecutive epochs e_i and e_{i+1}. In epoch e_i, AP slow down their apparent mining rate. This allows AP to mine blocks with lower difficulty value on C_A after difficulty recalculation in the epoch e_{i+1} and overtakes C_H. Note that, in this attack AP mines on C_A, whereas HP and RP mine on C_H.

Attack Strategy. In e_i, AP forks the blockchain to form a private chain C_A when r_1 fraction of e_i is completed (i.e., $r_1\lambda$ blocks are mined). AP creates blocks with timestamps such that the target recalculation leads to a very low difficulty for C_A in the next epoch e_{i+1}. Consequently, it can mine the blocks faster than HP and RP, and overcome C_H to become the longest chain. C_A overtakes C_H when $r_2 (\in (0, 1])$ fraction of total blocks in the epoch e_{i+1} (which is $r_2\lambda$ blocks) are mined.

Notice that AP does not need to mine the blocks slower. They have to put timestamps such that the blocks appear to be mined slower when made public. Further, broadcasted blocks must have timestamp < broadcasted time. Thus, while AP mines the blocks in epoch e_i slower than HPs (because $\beta_a < \frac{1}{2}$) due to the reduced difficulty of the private chain, it mines the remaining blocks of epoch e_{i+1} faster than HP and RP.

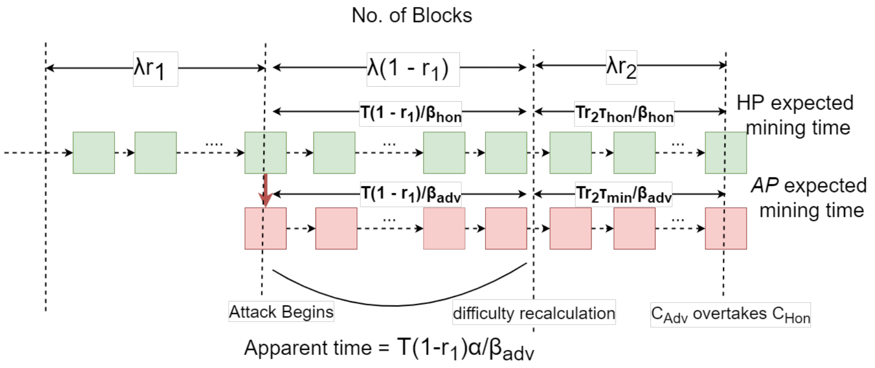

Figure 1: Difficulty Altering Attack

Analysis. Let T be the time to mine λ blocks of epoch e_i if all parties mine on C_H. If DIFFICULTY ALTERING is launched, let τ_{hon} be the difficulty adjustment for C_H and tau_{adv} for C_A. If DIFFICULTY ALTERING after difficulty-recalculation, in epoch e_{i+1}, $T\tau_{hon}$ is the time to mine all the blocks on chain C_H and $T\tau_{adv}$ is the time to mine on chain C_A. Note that, in this attack, $1 - \beta_a$ computing power is mining on C_H and β_a on C_A. Hence, all the expected required times need to be normalized accordingly.

The total time taken for C_A to complete r_2 fraction of e_{i+1} epoch should be $< C_H$, which gives us

$$\frac{T\tau_h r_2}{1-\beta_a} + \frac{T(1-r_1)}{1-\beta_a} > \frac{T(1-r_1)}{\beta_a} + \frac{Tr_2\tau_{adv}}{\beta_a} \tag{3}$$

In the equation above, the adversary slows down the apparent mining rate on C_A by a factor of α (> 1). Therefore, τ_{adv} and τ_{hon} are calculated as,

$$\tau_{adv} = max\left(\frac{1}{r_1 + \frac{\alpha(1-r_1)}{\beta_a}}, \tau_{min}\right) \qquad \tau_{hon} = max\left(\frac{1}{r_1 + \frac{(1-r_1)}{1-\beta_a}}, \tau_{min}\right)$$

Theorem 3.1 (DIFFICULTY ALTERING Attack)**.** *When $\tau_{min} < \frac{1}{2}$, an adversary can fork a PoW blockchain using* DIFFICULTY ALTERING *attack w.p. $> 1 - negl(\Theta\varepsilon)$ if $\beta_a \geq \underline{\beta_a}$.*

(i) expected time to mine a block by any party is $\propto \Theta$*,*
(ii) 2ε *is the difference in time between* AP *and* RP *mining the last block of the epoch* e_{i+1}*, and*
(iii) $\underline{\beta_a} = \frac{(3+\tau_{min})-\sqrt{(3+\tau_{min})^2-4(\tau_{min}+1)}}{2}$

Proof. The proof is in three steps. In Step 1, we state what environment the adversary sets to maximize its utility. In Step 2, we determine the fraction of computing power required by the adversary to launch the attack with the overwhelming probability, which we quantify in Step 3. The complete proof is in Appendix A.1. □

Corollary 3.1 (Bitcoin-DIFFICULTY ALTERING)**.** *Bitcoin is insecure against* DIFFICULTY ALTERING *Attack for* $\beta_a > 0.4457$.

The result follows from putting the value of $\tau_{min} = \frac{1}{4}$, as used in Bitcoin. Previously, bitcoin was considered secure against forking, with a high probability for $\beta_a < \frac{1}{2}$, however, we thus show that forking is possible even for $0.4457 < \beta_a < \frac{1}{2}$. One might argue this attack reduces the currency's price because it is a compromise in security, and $\theta = e_{security}$ after this attack. Therefore, any profit the AP gains in cryptocurrency is wasted. However, AP can profit from holding short position for the coin. We explain this incentive manipulation in detail in Appendix D.3

3.2 Rational-Byzantine attacks

The rational-Byzantine attack is when an AP launches an attack and relies on deviation from RP for the attack to be successful. We discuss two attacks, (i) QUICK FORK – security threat, and (ii) SELFISH MINING WITH BRIBING – fairness threat to the protocol.

3.2.1 Quick Fork Attack

In QUICK FORK, the adversary creates a fork $k(< \rho)$ blocks before the latest block of the longest chain. If the EO does not observe the fork as an attack, RPs are incentivized to mine on this forked chain for higher expected payoff. Since HP continues to mine on C_H, the deviating parties collect a larger fraction of the reward. Attacks discussed in Liao and Katz [2017a,b] are different from QUICK FORK since the QUICK FORK does not require abnormally large transactions to enable RP to deviate. It is not observable by EO, whereas the former attacks are easily observed by EO[11].

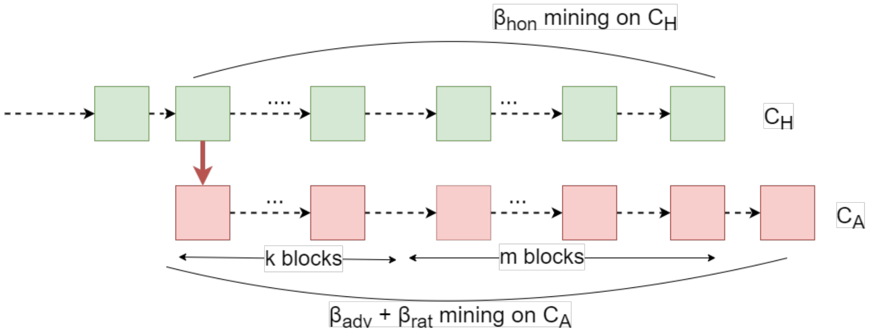

Figure 2: Successful Quick Fork Attack

Attack Strategy. Let C_H be the honest chain, and C_A be the forked chain. To launch this attack, the adversary has to ensure that at no time C_H has a lead of ρ or more blocks over C_A, where ρ is *tolerance factor* for EO. This ensures that the fork is not considered an attack by the EO, thereby incentivizing RP to mine on the forked chain as they may grab more block rewards than mining on the C_H.

Definition 3.1 (QUICK FORK)**.** We say the adversary has launched QUICK FORK successfully if (i) it forks the honest chain C_H at $k < \rho$ blocks previous to the latest block on C_H creating C_A, (ii) C_A becomes the longest chain, and (iii) C_H does not exceed C_A by $> \rho$ blocks during the attack.

[11] eg. transaction – cc455ae816e6cdafdb58d54e35d4f46d860047458eacf1c7405dc634631c570d

If $k \geq \rho$, RP do not mine on the forked chain C_A, as C_A is observed as an attack by the EO and conversion rate $\theta(t)$ falls to $e_{security}$. Due to this, RPs do not shift to the C_A, which therefore does not overtake C_H. Therefore, QUICK FORK is possible only at $k < \rho$ blocks from the end of the longest chain.

Analysis. We show that the QUICK FORK attack strategy is dominant over Π for AP and RP. We show conditions under which QUICK FORK attack is possible with high probability.

Theorem 3.2 (QUICK FORK Attack)**.** *Adversary successfully launches* QUICK FORK *with probability of* $\frac{n-1}{n}$ *if*

- $\eta > \frac{n-\beta_h n-1}{n\beta_h - 1}$
- $\beta_a + \beta_r \geq \frac{1}{2}$
- $\beta_h > \frac{\chi}{r_{block}}$
- $k < \rho - \varpi$

Here, the cost incurred by the system on mining one block is χ_1. r_{block} *is the block-reward for the current phase* P_i *and* $\eta = \frac{r_{block}}{\chi_1}$, $\phi = \frac{\beta_h}{\beta_a+\beta_r}$ *and* $\varpi = log_\phi(\frac{1+(n-1)\phi^\rho}{n})$

Proof. In the proof, we compare payoff for RP, and AP in following and deviating from the protocol from C_H state of the blockchain, from when the fork happens, till C_A overtakes C_H. The proof goes into four steps. In Step 1, we compute the payoff on deviation; in Step 2, we calculate the payoff on following the protocol. In Step 3, we compare the results from Step 1 and Step 2 and show the conditions that make QUICK FORK feasible. In Step 4, we argue about the probability which the attack is successful. The complete proof is provided in Appendix A.2. □

3.2.2 Selfish Mining with Bribing attack

Eyal and Sirer Eyal and Sirer [2014], showed that with computing power of $\frac{1}{4}$, (i.e., $\beta_a = 0.25$ or higher), the attack *selfish mining* (described in Section 2.3.1) is feasible. In this attack, AP mines block without revealing to the network (called private chain) and announce its private chain at appropriate times (refer to Eyal and Sirer [2014] for more details), grabbing more rewards than its fair share, leading to $\theta(t) = e_{fairness}$. The authors assume that all miners are either in AP (part of the pool launching a selfish mining attack) or HP. In our model, miners are in either AP, RP, or HP. For such realistic scenarios, we prove that the attack is possible even if $\beta_a < \frac{1}{4}$. The key intuition is that the adversary can bribe RPs to switch to its chain (C_A), thereby increasing the success probability for the attack. In summary, the $\frac{1}{4}$ bound in Eyal and Sirer [2014] is a special case of a more general bound on β_a (Theorem 3.3) for feasibility of Selfish Mining attack.

Therefore, our contribution is to show that if we consider the system to have HP, AP and RP, then the bound on selfish-mining is actually more general if we introduce bribing in Selfish mining.

Attack Strategy. The attack model is similar to that discussed in Eyal and Sirer [2014]. The only difference is that in each block that the AP mines (privately), it includes a *bribe transaction*[12]. The modified protocol proposed in Eyal and Sirer [2014] mandates the parties to choose one of the competing blocks randomly and mine on top of the chosen block. This choice is not known to other parties or the EO. Thus, RP can choose to mine on top of the Adversarial Block to collect bribes without EO realizing the deviation.

Analysis. The total mining power mining on top of a non-adversarial block is $\frac{\beta_h}{2}$ (because each miner in HP chooses randomly one of C_H or C_A to mine on), while on an adversarial block, a total of $\frac{\beta_h}{2} + \beta_r + \beta_a$ mining power is mining. Therefore this chain wins the race (block is mined) with a higher probability of $\frac{\beta_h}{2} + \beta_r + \beta_a$. The analysis leads to Theorem 3.3, which is proven in Appendix A.3.

Theorem 3.3 (SELFISH MINING WITH BRIBING Attack)**.** *Attack strategy* SELFISH MINING WITH BRIBING *is dominant over the Honest protocol for*

$$\beta_a > \underline{\beta}^{SMB} = \frac{\beta_h}{2\beta_r + 4\beta_h} \tag{4}$$

Here $\underline{\beta}^{SMB}$ *is the lower bound on* β_a *for which* SELFISH MINING WITH BRIBING *attack is profitable.*

From Theorem 3.3, we derive bound on β_a for SELFISH MINING WITH BRIBING attack.
Case 1 : (Fully Rational Setting) Here, $\beta_h = 0$ and therefore $\gamma = 1$. The attack is profitable for $\beta_a > 0$.

[12] easy to grab a small reward whoever mines on the adversarial block.

Case 2: (Mixed Setting) In this setting $\beta_h > 0$, therefore, the bound when this attack is feasible for the attacker becomes $\beta_a > \underline{\beta}^{SMB}$.

By trivial analysis, we see that $\underline{\beta}^{SMB} \leq \frac{1}{4}$ (equality when $\beta_r = 0$). However, the actual $\underline{\beta}^{SMB}$ can be much smaller depending on β_r. The gist of the analysis is that selfish mining is possible with much less computing power than the previously known bound of 25% in the presence of rational parties.

3.3 Rational attacks

Rational Attacks are deviations from the original protocol Π or some sub-routine of Π. In this case, each rational agent deviates from the protocol, and no adversary is involved. We discuss one such deviation from the *Gossip* sub-protocol of Π, the transaction withholding attack.

3.3.1 Transaction Withholding attack (σ_{tw}

When a miner receives a transaction, they add it to the mempool and share it with their peers, which is the expected behavior of Gossip protocol Π_{Gossip}. In Block-Reward Model (BRM) RP has a negligible incentive to deviate from Π_{Gossip}. In the Transaction Fee Only Model, such deviations are incentivized, as shown by Theorem 6 in Badertscher et al. [2018]. We show one such deviation – transaction withholding attack (represented as σ_{tw}) which dominates Π_{Gossip} as shown in Lemma 3.4.

Lemma 3.4. *In TFOM, for any rational party, the following σ_{tw} strictly dominates Π_{Gossip}.*

Proof. We first calculate the payoff for a RP in the following Π_{Gossip} to prove this. We then calculate the payoff on the following σ_{tw} and show that the latter payoff is strictly greater than the former. We provide the complete proof in Appendix A.4. □

This deviation poses one (or both) of two possible threats:

- Low Throughput: The chance of a single RP mining a block is minimal; therefore, the transactions take a long time to get accepted in the blockchain, thus reducing the throughput.
- Centralization: The protocol becomes centralized if the transaction proposer sends the transaction to a large mining pool in the hope of getting the transaction published quickly.

Where do previous frameworks fail?

We discussed four possible attacks in PoW blockchains. We shall discuss why previous models (most specifically Rational Protocol Design (RPD)) could not capture these deviations. Three properties were not captured by previous works and are discussed below.

3.3.2 Agent Modelling

Some miners could be rational, i.e., want to maximize their utilities without disrupting the protocol. In Badertscher et al. [2018], Protocol Descriptor's utility comprises of honest miner's utility and a very high negative payoff for events such as forking ($exp(polylog(\kappa))$). It fails to capture the objective of the Protocol Descriptor – maximize the difference between parties that follow protocol and parties that deviate. Thus, RPD could not capture deviations discussed in Section 3.2.

3.3.3 Externalities

In RPD, the conversion rate of the underlying crypto-currency to fiat currency is considered constant. However, the loss of utility due to security attacks should be calculated through a change in *externality*, which affects $\theta(t)$. Modeling such externality allows quantifying the loss in utility for AP and RP on deviating from the protocol.

3.3.4 System Dynamics

RPD and its derivatives do not account for (i) block rewards changing over time and (ii) variable difficulty of mining. Because of this, previous models could not capture attacks discussed in Section 3.1.

4 pRPD for Blockchains

In this section, we introduce the Practical Rational Protocol Design(pRPD]), an improved model for the game-theoretic security analysis of PoW blockchains. pRPD models blockchain as a two-player Stackelberg game between Protocol Descriptor and Adversary.

4.1 Players

There are two players in our modeling of the protocol as a game: (i) Protocol Descriptor and (ii) AP.

- **Protocol Descriptor (PD)**: The protocol descriptor is the player who selects the protocol Π which will be considered as the honest protocol. PD must ensure that (i) The relaxed ideal-functionality $\mathcal{G}_{weak-ledger}$ is realized by this functionality, and (ii) This protocol gives them the maximum utility.
- **Adversarial Party (AP):** We exploit the notation to use AP to refer to both the set of adversarial miners and the adversary who is part of the two-player game. This is because the set of adversarial miners is controlled by a PPT ITM $\mathcal{A}$, which carries out the adversarial strategy. This $\mathcal{A}$, which decides the adversarial strategy given a protocol Π, is the AP in this game.

4.2 Attack Model

As defined in Garay et al. [2013], Badertscher et al. [2018], we need to define an attack model for game-theoretic analysis of the protocol's security. The attack model is parameterized by the tuple $(\mathcal{F}, <\mathcal{F}>, v_A, v_D)$. In this, $\mathcal{F}$ is the ideal functionality that the protocol wishes to realize in the real world. $<\mathcal{F}>$ is the relaxation of the ideal functionality. In the case of PoW Blockchain protocols, $\mathcal{G}_{ledger}$ is the ideal functionality, and the $\mathcal{G}_{weak-ledger}$ functionality as its relaxation.

Further, v are mappings $v : \mathcal{S} \times \mathcal{Z} \longrightarrow \mathbf{R}$ from the simulator, simulating attack strategy $\mathcal{A}$ on protocol Π, in environment $\mathcal{Z}$ to real-valued payoff. The utility is defined as the expectation over this payoff function. In our case, the three types of parties HP, RP and AP have payoff vectors v_H,v_R, and v_A respectively. Therefore, our attack-model is

$$\mathcal{M} = (\mathcal{G}_{ledger}, \mathcal{G}_{weak-ledger}, v_A, v_R, v_H)$$

4.3 Game

Typically, a blockchain system consists of two types of participants (i) protocol descriptor and (ii) miners – we call each type of miner as *parties* (already described in Section 2.2.3). Additionally, we assume an *external observer* who is not part of the game. The game progresses as follows:

❒ The Protocol Descriptor defines the protocol. Its role is to choose the best protocol Π, which maximizes its utility (and, by design, of the players who follow Π).

❒ AP observes Π and chooses an attack strategy implemented by any ITM $\mathcal{A} \in \mathcal{C}_A$ for the choosen Π.

We model the interaction of the adversary parties with the system as a two-player Stackelberg game between Protocol Descriptor and the adversary where the leader of the game is Protocol Descriptor, and the follower is AP. In the system, there is EO who determines the conversation rate $\theta(t)$ – the price of one unit of underlying cryptocurrency to a fiat currency at round t. Our game. $\mathcal{G}_{\mathcal{M}}$, is defined over attack-model $\mathcal{M} = (\mathcal{G}_{ledger}, \mathcal{G}_{weak-ledger}, v_H, v_R, v_A)$. The RP and HP are not part of the Stackelberg game because for HP strategy is fixed Π, when Protocol Descriptor moves, and for RP the most optimal out of the possible deviations is fixed after AP selects their strategy.

4.3.1 Security Definitions

Similar to Badertscher et al. [2018, 2021] say the 'best possible behavior' that is practically achievable for the adversary is a semi-honest front-running strategy. The front-running semi-honest adversary A_{fr} is a subset of front-running adversary A^*_{fr}.

Definition 4.1 (Front-Running Semi-Honest Adversary (Def. 2, Badertscher et al. [2018])). An adversary $\mathcal{A}$ which is in the set of adversaries $\mathcal{A}_{fr}$ is said to be semi-honest, front-running if

- upon activation, the adversary corrupts miners and follows Π (honest protocol).
- any message that the adversary wants to broadcast, it does so immediately

- if a non-adversarial party wants to broadcast the message, the adversary maximally delays that message by one round.

Now that we have defined the behavior of the adversary we want to achieve, we define the strongest achievable security guarantee for a protocol: strong attack-payoff security. We motivate the definition of *strong attack-payoff security* from Badertscher et al. [2018].

Definition 4.2 (Strong Attack-Payoff Secure (Def. 3, Badertscher et al. [2018])). A protocol Π is *strongly attack-payoff secure* under attack-model $\mathcal{M}$ if for some adversary in the set of semi-honest, front-running adversary $A \in \mathcal{A}_{fr}$, the attacker playing A is approximate best response strategy. That means, $\forall A_2 \in$ PPT ITM and $A_1 \in \mathcal{A}_{fr}$,

$$U_A(\Pi, A_2) \leq U_A(\Pi, A_1) + negl(\kappa)$$

4.4 Utility

For calculating the utility, we define random variables for the payoff for each $\mathcal{A}, \mathcal{R}$, and $\mathcal{H}$. The random variable $v_{\mathcal{X}}$ for $\mathcal{X} \in \{\mathcal{A}, \mathcal{R}, \mathcal{H}\}$ is defined over environment $\mathcal{Z}$. On the lines of the Universal Composability paradigm, since the game is a Stackelberg Game, suppose that the Protocol Descriptor chooses strategy Π, and the AP chooses $\mathcal{A} = A(\Pi)$. In that case, let $\mathcal{S}$ be an ideal-world simulator for $\mathcal{A}(\Pi)$, where $\mathcal{S}$ attacks on $< \mathcal{F} >$. The set of all such simulators is $\mathcal{C}_A$.

4.4.1 Objective of the Model

The AP's objective is to achieve more payoff for itself. Further, we observe that, for AP and EO (both PPT ITMs), the HP and RP are indistinguishable from each other. Thus, the objective of the protocol descriptor is to achieve more payoff for HP and RP and less for the AP. Notice that it is at this point that the utility model differs from most of the previous works. It also captures the dynamic nature of the protocol by considering payoff across different rounds with changing protocol parameters.

Previous utility models captured only deviations which benefit the adversary. However, AP can gain higher utility in the long run by reducing the payoff of HP. We elaborate on how this is possible in Appendix D.2. The payoff of Protocol Descriptor is thus the difference between the payoff of non-deviating parties HP and RP (in view of EO) and deviating party AP. Another distinction from RPD is how externalities are modeled in the payoff of the miners. In our work, we account for critical security threats and moderate fairness threats through externality (reflected by the conversion rate $\theta(t)$). This term decreases both AP and Protocol Descriptor payoff. However, AP payoff can increase if they are holding short position against the cryptocurrency. To model such situations, while calculating U_A, we add an extra term in v_A, which is inversely proportional to the currency's conversion rate.

4.4.2 Utility Model

Consider the attack model as described in Section 4.2. If an attack $A(\Pi)$ is defined on a chosen protocol Π, let $C_{\mathcal{A}}$ be the set of simulators, which simulate the attack on the relaxed, ideal functionality $< \mathcal{G}_{weak-ledger} >$. Given $\mathcal{S} \in C_{\mathcal{A}}$ and environment $\mathcal{Z}$, v_H, v_A, v_R are expected payoff of HP, AP, and RP. We find the normalized payoff of a single miner as $\frac{v_H + v_R}{n(t)(\beta_{Hon} + \beta_{Rat})}$ for single honest (non-deviating for EO) miner and $\frac{v_A}{n(t)\beta_{Adv}}$ for single AP (deviating w.r.t. EO). If we multiply the utility model with a positive constant, we use the fact that the players' best strategies do not change and drop $n(t)$ – the number of miners in the system at round t.

For AP, we minimize over this set $C_{\mathcal{A}}$ for the choice of simulator $\mathcal{S}$. This is because $C_{\mathcal{A}}$ contains all the simulators which can launch the attack. Many of these may invoke additional events unrelated to the attack. However, the purpose of AP is to force the simulator to invoke the attack in the ideal world. Hence, the most closely related payoff is of the simulator that "just" simulates the attack in the ideal world. Hence, we minimize over the set of all simulators, $\mathcal{C}_{\mathcal{A}}$ for AP. For Protocol Descriptor's utility, which is a function of the expected payoff of HP, RP and AP, we consider the worst environment ($\mathcal{Z}$), and for AP's utility, the best environment. In summary, utilities for a given adversarial strategy $A(\Pi)$ for Protocol Descriptor strategy Π is as follows.

$$U_{\mathcal{D}}^{\Pi,<\mathcal{F}>}(A) = \min_{\mathcal{Z} \in ITM} \left\{ \min_{\mathcal{S} \in C_{\mathcal{A}}} \left\{ \frac{v_{\mathcal{H}} + v_{\mathcal{R}}}{\beta_{Hon} + \beta_{Rat}} - \frac{v_A}{\beta_{Adv}} \right\} \right\} \tag{5}$$

$$U_{\mathcal{A}}^{\Pi,<\mathcal{F}>}(A) = \max_{\mathcal{Z} \in ITM} \left\{ \min_{\mathcal{S} \in C_{\mathcal{A}}} \left\{ \frac{v_A}{\beta_{Adv}} \right\} \right\} \tag{6}$$

Our adversary is an $\mathcal{M}-$*maximizing adversary*. This means that given a protocol Π chosen by the Protocol Descriptor, the adversary chooses the best response attack A, which maximizes their utility function $U_A(\cdot)$.

Definition 4.3. An adversary is $\mathcal{M}-$maximizing adversary if given protocol Π which realizes functionality $< \mathcal{F} >$, they choose strategy such that for their utility function $U_{\mathcal{A}}^{\Pi,<\mathcal{F}>}(\cdot)$ is maximized.

$$U_{\mathcal{A}}^{\Pi,<\mathcal{F}>} = \max_{A\in ITM} U_{\mathcal{A}}^{\Pi,<\mathcal{F}>}(A)$$

4.4.3 Advantages of Our Utility Model

The advantages of our utility model are as follows:

1. It models the *externality* more flexibly, allowing an EO who can observe certain types of deviations as attacks and reduce the $\theta(t)$ correspondingly. This also allows us to model market responses differently to different type of attacks.
2. It can represent *variable block reward.* Further, it does not restrict to a specific type of reward model[13], but a general series that can converge or diverge.
3. It captures *variable difficulty* because the probability of mining in each round is different from each other.
4. The *utility* of the protocol descriptor does not just try to increase the payoff of HP but also decreases the difference between HP and AP utility. This is better than previous models because it does not allow high utility for such attacks, which increases the HP utility but increases the AP utility even further (possibly due to external payoffs).

5 Detering Attacks

In this section, we propose modifications to the original PoW blockchain protocol that helps us tackle the attacks discussed in Section 3.

5.1 Difficulty Altering Attack

From Theorem 3.1, it is clear that we can overcome possibility of DIFFICULTY ALTERING attack if $\underline{\beta_{adv}}$ is at least $\frac{1}{2}$. If $\beta_{adv} > \underline{\beta_{adv}} = 0.5$, then no PoW blockchain is secure. In this subsection, as a corollary to Theorem 3.1, we show that we can achieve this if we appropriately set τ_{min}.

Corollary 5.1. *A PoW blockchain with $\beta_{adv} < \frac{1}{2}$ is secure against* DIFFICULTY ALTERING *Attack if $\tau_{min} \geq \frac{1}{2}$*

Proof. From the proof of Theorem 3.1, lower bound on β_{adv} to launch DIFFICULTY ALTERING Attack is $\underline{\beta_{adv}} = \frac{(3+\tau_{min})-\sqrt{(3+\tau_{min})^2-4(\tau_{min}+1)}}{2}$. We want $\underline{\beta_{adv}} \geq \frac{1}{2}$. With simple algebra, one can argue that $\underline{\beta_{adv}} \geq 0.5$ if $\tau_{min} \geq \frac{1}{2}$.[14] □

5.2 QUICK FORK Attack

To defend against QUICK FORK attack, we exploit that RP and HP are indistinguishable for AP. For RP to mine on adversarial chain (C_A) with the help of RP, AP makes C_A public making it visible to HPs too. We propose that all parties (non-deviating) are allowed to add *Proof-of-Invalidity* (PoI) to any chain shorter than the chain on which they are mining. With POI, we can prevent QUICK FORK attack.

PC-MOD **Solution** We propose that blockchain protocols allow party par mining on chain C_{par} at height a_{par} to add a block containing POI on a forked chain (C_A) if it observes C_A is of height $\leq a_{par} - k_{th}$. In this, k_{th} is the *threshold gap*, which is defined below.

Definition 5.1 (Threshold Gap)**.** Threshold Gap (k_{th}) is the difference in the height of the longest chain(C_H) and forked chain (C_A) such that $\beta_{rat} + \beta_{adv}$ mining on C_A can overtake C_H w.p. $\geq 1 - \mu$.

$$k_{th} = \rho - log_{\phi}(\mu + \phi^{\rho}(1 - \mu)) \tag{7}$$

Here, $\phi = \frac{\beta_{hon}}{1-\beta_{hon}}$ and ρ is the tolerance factor of the EO. This relation is derived by following a similar argument as in Step 4 of Theorem 3.2 and results from Section 4.5.1 of Ross [1975]

[13] most of the previous works have stuck to constant block-reward

[14] we do not discuss $\beta_{adv} > \frac{1}{2}$ because forking is possible in that case by 51% attack.

We want PoI should satisfy two conditions. (1) A block containing that POI is indistinguishable from any other block; otherwise, the RP and AP ignore that block and mine on top of its parent block, and (2) Adversary should not be able to add POI on the honest, longest chain and invalidate that chain. We construct PoI from the definition below, which satisfies the two requirements.

Definition 5.2 (Proof of Invalidity)**.** The PoI *transaction* is published on the forked chain to prove its invalidity. It is constructed as:

- PoI consists of a string h which is the hash $H(H_{a_{par}}||m_{secret})$[15]. Here $H_{a_{par}}$ is the hash of the block at height $\geq a_{par}$ on the C_H, and m_{secret} is a secret string chosen by the proposer of PoI.
- Since h is an arbitrary string, this transaction is indistinguishable from any other transaction.
- Since the proposer of the invalidity transaction is HP, they can invalidate the chain C_A if it overtakes the C_H as the longest chain by revealing the m_{secret}. If the PoI is added to C_A at height $\leq a_{par} - k_{th}$, the PoI is considered valid.

Claim 5.1. *The probability of HP being successful in adding POI on C_A is $\geq 1 - e^{-\beta_{hon} \cdot k_{th}}$.*

Proof. Initially, C_A trails behind C_H by k_{th} blocks. HP can add POI in these k_{th} blocks only because the height at which POI is present should be less than the height of the block whose hash it contains (which is at height a_{hon}). Therefore, if an adversary mines a block among these k_{th} blocks, they can successfully add POI to that block. Let E_{mit} be a chance that HP successfully mines a block in these k_{th} blocks. Using the inequality $e^{-x} > 1 - x$ we get $P[E_{mit}] = 1 - (1 - \beta_{hon})^{k_{th}} > 1 - e^{-\beta_{hon} k_{th}}$ □

Observe that the above probability only accounts for HPs adding POI via mining a block. In practice, HP can broadcast the POI transaction, and all parties mining on C_A add the transaction as it is indifferent from any other transaction. Thus, the actual probability is higher than in Claim 5.1.

Theorem 5.1. *PoW blockchain protocol with PC-MOD*
(i) it is an equilibrium for RPs to mine on the longest chain and not to shift to C_A, the forked chain in QUICK FORK *attack. (ii) Protocol is secure against* QUICK FORK *attack for $\beta_{adv} < \frac{1}{2}$ with high probability.*

Proof. The PC-MOD modification mandates HPs to mine on the C_A unless their POI transaction is included in one of the blocks in the chain. This makes the deviation to launch (and join) QUICK FORK Attack disincentivized due to three reasons:

1 All $\beta_{adv} + \beta_{rat} + \beta_{hon}$ parties mine on C_A, so any advantage that the RP or AP might have gotten due to increased share of block-reward is now not present.

2 RP is disincentivized to mine on C_A as (i) the block reward is not higher than mining on the longest chain C_H, and (ii) if RP shifts to mining on C_A, the mining cost on C_H between on $C_H - C_H^{\lfloor k^{th}}$ is wasted, implying lesser utility than mining on C_H.

3 If such an attack still takes place, there is always the risk of HP mining a block or a valid POI transaction (indistinguishable from other transactions) is included in the C_A, which leads to $\theta(t) = e_{security}$. This happens with probability $> (1 - (1 - \beta_{hon})^{k_{th}})$ thus disincentivizing the RP from participating in the attack.

Thus, for $\beta_{adv} < \frac{1}{2}$, the attack happens only if β_{adv} can fork the chain by itself, which is possible with negligible probability. □

5.3 SELFISH MINING WITH BRIBING Attack

We show a rather pessimistic result in case of SELFISH MINING WITH BRIBING attack. This result shows that it is impossible for a protocol which realizes the ledger functionality $\mathcal{G}_{weak-ledger}$ to be resilient to SELFISH MINING WITH BRIBING attack.

Theorem 5.2 (Selfish-Mining Impossibility)**.** *For any PoW-Blockchain which UC-realizes $\mathcal{G}_{weak-ledger}$, the protocol can't be* strongly attack-payoff secure *because* SELFISH MINING WITH BRIBING *is always possible if front-running is possible.*

[15] here H is the hash-function used in PoW blockchain.

$\Pi_{Tx-Inclusion}$

```
Input: $dest$[a], $tx$[b], $\mathfrak{H}$[c]
if at last $l$ bits $\mathbf{H}(tx, \mathfrak{H}) = \mathbf{H}(dest, \mathfrak{H})$ return True
else return False
```

[a] miner's destination address
[b] transaction
[c] hash of parent block

Proof. The proof follows as a direct result of the Lemma 5.3, which is stated below. If for every protocol, each honest execution has an indistinguishable SELFISH MINING WITH BRIBING counterpart, then for every such protocol, SELFISH MINING WITH BRIBING attack is possible. Note that the proof, as standard in the literature Badertscher et al. [2018, 2021], inherently assumes that the best response A for any adversary is front running, i.e., $A \in \mathcal{A}_{fr}$. □

Lemma 5.3. *For a protocol Π, there exists an environment Z_1 and a simulator for front-running adversary $S_1 \in \mathcal{A}_{fr}$ and a corresponding environment Z_2 and an simulator for adversary using* SELFISH MINING WITH BRIBING $S_2 \in \mathcal{A}_{SMB}$ *such that for any PPT observer, execution (S_1, Z_1) and (S_2, Z_2) are indistinguishable.*

Proof. Proof is provided in Appendix B.1 □

5.4 $\Pi_{tx-inclusion}$ Protocol

To resolve the Transaction withholding attack (Section 3.3), we propose a modification in the form of an additional sub-protocol over the Π_{gossip}. This sub-protocol ($\Pi_{Tx-Inclusion}$) is a filter by which each miner can add only transactions satisfying a certain condition in the current block. With this modification, we can argue that RP's gain in utility by withholding transaction is negligible, implying that following Π_{gossip} is approximate Nash-equilibrium over the transaction withholding deviation.

5.4.1 Proposed Modification

A RP mining a block can add only those transactions in the block which satisfy condition **C1**.

C1: A transaction tx satisfies **C1** given the block and coin-base address $dest$ (similar to Pay2PubHash in Bitcoin En.bitcoin.it [2021]) with parent block hash $\mathfrak{H}$ if the last l bits of $\mathbf{H}(tx, \mathfrak{H})$ and $\mathbf{H}(dest, \mathfrak{H})$ are same. If coin-base is a script (ex. Pay2ScriptHash in Bitcoin En.bitcoin.it [2017]), $dest$ is the script hash.

To add as many transactions as possible, HPs may need to maintain multiple keys for which we can use PKI Trees (Buldas et al. [2017]) which takes logarithmic space for key storage. With these modifications, theprobability of a party mining a block and simultaneously including the withheld transaction is reduced because of one of two reasons:

1 If the party randomly selects an address and spends all the computing power on PoW for mining the block, there is a $\frac{1}{2^l}$ chance of that transaction being valid to be in the block.

2 If the party spends some of its mining power on finding a favorable address mapping, then the number of queries they can perform for PoW reduces, thereby reducing their probability of mining a block. Also, the address mapping created by the party is not useful for the next round.

Lemma 5.4. *If $\Pi_{tx-inclusion}$ is followed, $\Pi_{gossip}$is ϵ_G-Nash Equilibria, for $\epsilon_G = tx \cdot O(2^{-l})$. Here, tx is the cumulative fee from the transaction sent to the party and is poly in l.*

In summary, on following $\Pi_{tx-inclusion}$, TRANSACTION WITHHOLDING attack gives no significant payoff as shown in Lemma 5.4. The proof follows by computing the difference in the payoff of following and deviating from Π_{gossip}, which is $\leq \epsilon_G$. We provide the calculation in Appendix B.2.

6 PRAGTHOS & Theoretical Analysis

We have discovered multiple attacks on PoW Blockchains (which also exist in Bitcoin). The previous game-theoretic analysis Badertscher et al. [2018], Garay et al. [2013], Karakostas et al. [2022], Judmayer et al. [2020] primarily focused on static population and horizon in which block-rewards and difficulty are constant. Our analysis framework, proposed in Section 4, is very general and could discover the before-mentioned attacks (Section 3). In Section 5, we proposed novel solutions to these attacks by (1) Proposing additional sub-protocols in the PoW blockchain or (2) Specifying hyper-parameter values. With these modifications, we abstract out a new framework for PoW blockchain protocols. We call it **PRAGTHOS**, – Practical Rational Game Theoretically Secure. It also is a conjunction of words *'Pragmatic'* meaning logical (rational), and *'Ethos'*, which roughly translates to character, describing the Rational Characteristic of the users of the protocol. In this section, we first summarize **PRAGTHOS**, and then (Section 6.2) provide its security analysis.

6.1 Modification to PoW Blockchain

PoW Framework As mentioned in Section 2.2, in a PoW blockchain, parties mine a block by solving a cryptographic puzzle. The puzzle encompasses finding a random nonce along with the merkle root of transaction data, header data is fed to hash again, and the final hash should be less than a certain target determined by difficulty recalculation at the start of each epoch. The parties are expected to collect all transactions they hear and adjust difficulty at the end of the epoch to maintain the average duration between two blocks as same as possible. The ratio of the previous difficulty and the new difficulty must be $\in [\tau_{min}, \tau_{max}]$. The block rewards change by a factor ϑ across phases. Let the block-reward scheme followed by the protocol be given as $g(0), g(1), \ldots$ where $g(i) = r_{block}(0) \cdot \vartheta(i)$.[16] denotes the block reward in Phase i. For bitcoin, $\vartheta(i) = \frac{1}{2^i}$.

When the sequence $< \vartheta >$ is converging (i.e. $\sum_{i=0}^{L} \vartheta(i)$ is finite for all L), the underlying crypto-currency is called *deflationary*; otherwise we call it *inflationary*.

Definition 6.1 (Inlfationary Crypto-Currency)**.** We say, a PoW crypto-currency is *inflationary* if block-rewards update according to $r_{block}^{new} = \varpi(r_{block}^{old}, i) = r_{block}^{old} \frac{\vartheta(i)}{\vartheta(i-1)}$, and $< \vartheta >$ is diverging[17].

With the modifications stated in Fig. 3 to PoW protocols, **PRAGTHOS** is strongly attack-payoff secure if $\beta_{adv} < \frac{1}{2}$ and ensures fairness (against SELFISH MINING WITH BRIBING attack) if $\beta_{adv} < \underline{\beta}^{SMB}$.

In **PRAGTHOS**, the PoW blockchain undergoes the following modifications.

❏ *PC-MOD.* All parties are expected to add POI (Definition 5.2) if they observe a fork that is at least k_{th}(Eq. 7) block behind their current chain.

❏ *Difficulty Adjustment.* To protect against DIFFICULTY ALTERING Attack, the parties are expected to use $\tau_{min} = \frac{1}{2}$ while updating difficulty at the end of each epoch.

❏ *For Adding Transactions.* For collecting transactions in a block, it follows $\Pi_{Tx-inclusion}$.

Figure 3: Pragthos Framework

First, we need to assume that if every miner is honest, the reward structure is such that mining is profitable, compensating the costs incurred. We call it *All-honest-profitability*. This condition ensures for all HP the protocol is *Individually-Rational*[18]. Note that, we are not assuming $\beta_H = 1$ in the analysis.

Definition 6.2 (All-honest-profitability)**.** We say a PoW blockchain block-reward scheme satisfies *All-honest-profitability* at round t if for a system where $\beta_H = 1$ we have $\theta(t) r_{block}(t) p_H > \chi(t)$. Here, p_H is the probability of a single miner mining a block in round t.

6.2 Results

PoW blockchains can be forked by AP having majority computing power (through 51% attack), due to which mining need not be profitable for HP. Thus, we assume that $\beta_{adv} \leq \frac{1}{2}$. However, as indicated in Section 3.1, even with this, in a typical PoW, blockchain is susceptible to attacks which might lead to $\theta(t) = e_{security}$, making mining not profitable.

[16] this relation can also be written as $r_{block}^{new} = \varpi(r_{block}^{old}, i) = r_{block}^{old} \frac{\vartheta(i)}{\vartheta(i-1)}$

[17] diverging $\Rightarrow \lim_{L \to \infty} \sum_{i=0}^{L} \vartheta(i) \longrightarrow \infty$

[18] Individual-rationality means payoff from participating in the protocol is $\geq$ the payoff from abstaining from participating.

6.2.1 Strong Attack-Payoff Security for Inflationary Currency

In this section, we analyze and prove in Theorem 6.1 **PRAGTHOS** is strongly attack-payoff secure (Definition 4.2) under an inflationary block-reward scheme (sufficiency condition). We further prove that such inflation in **PRAGTHOS** is *necessary* for any PoW blockchain protocol to be strongly attack-payoff secure (Theorem 6.3).

Theorem 6.1 (Strong attack-payoff Security – Sufficiency)**.** ***PRAGTHOS*** *is* strongly attack-payoff secure *under* $\beta_{adv} < \frac{1}{2}$ *if reward scheme is inflationary and satisfies All-honest-profitability.*

Proof. To prove the result, we leverage the UC framework, originally developed by Canetti Canetti [2001], further illustrated for blockchains by Badertscher et al. Badertscher et al. [2017]. We briefed it in Section 2.1. With this, the proof directly follows from Lemma 6.2, as **PRAGTHOS** satisfies all three conditions (C1-C3) of the Lemma. □

Lemma 6.2. *Let A_{fr} be the class of semi-honest, front-running adversaries. For each adversarial strategy A_2, these exists adversarial strategy $A_1 \in A_{fr}$,*

$$U(\Pi, A_1) + negl(\kappa) \geq U(\Pi, A_2)$$

and it is true when the following are satisfied:

C1 *Reward-scheme and externality is such that it satisfies* All-honest-profitability.

C2 $\beta_{adv} < \frac{1}{2}$

C3 *The block-reward scheme is inflationary.*

Proof. This proof proceeds in 3 steps (7 sub-steps). In Step 1, we find the utility of a front-running adversary $A_1 \in A_{fr}$. More specifically, we find the environment under which this adversary exists and the Reward $\mathcal{R}_{A_1}$ in Step 1a. Then in Step 1b, we find an appropriate lower bound on the probability of mining by A_1, after considering the variable difficulty and dynamic population. Finally, in Step 1c, we take into account variable block reward (inflationary) and find a lower bound on expected reward for A_1, or $E[\mathcal{R}_{A_1}]$.

In Step 2 of the proof, we upper bound the payoff of any other arbitrary adversary A_2 for its maximizing environment Z_2. In this case, we find the upper bound on the expected payoff of the adversary A_2. Then in Step 3a, we argue that an environment Z_1 always exists for every Z_2, such that a condition holds true. We argue that under such an environment, except with negligible probability, the payoff of A_1 exceeds the expected payoff of A_2. Finally, in Step 3b, we argue by contradiction that the adversarial setting (S_1, Z_1) is strongly attack-payoff secure. Where, S_1 is the ideal world simulator of $A_1 \in A_{fr}$. (ref.Appendix C.1 for complete proof) □

The *all-honest-profitability* condition is to ensure non-deviating parties participate in the system. Further, AP has incentives both internal (through coins), and external (through short position on the currency) and is therefore incentivized to participate irrespective of *all-honest-profitability* condition. Since we proved this theorem for general diverging series ϑ, this is true for series such as constant-series ($\vartheta(i) = c$), harmonic series ($\vartheta(i) = \frac{1}{i}$) etc.

6.2.2 Results for Deflationary Currency

One of the advantages of **PRAGTHOS** is that even under a deflationary reward scheme, it provides strong attack payoff security against a class of adversaries whose attacks are bounded by the number of rounds. In this section, we first show that PoW blockchains with geometrically decreasing block-reward schemes (like Bitcoin) are not strongly attack-payoff secure against a PPT ITM adversary. We then show that such a PoW blockchain when following **PRAGTHOS** framework, is strongly attack-payoff secure against a PPT ITM adversary with an upper limit on the number of rounds on their attack.

Theorem 6.3 (Deflationary Reward Scheme)**.** *PoW blockchain with geometrically decreasing Deflationary Reward Scheme, ($\vartheta(i) = \vartheta^i$ for $\vartheta < 1$) cannot be strongly attack-payoff secure in Block-Reward model. We assume the protocol initially (at $t = 0$) satisfies* all-honest-majority.

Proof. This proof follows in three steps. In Step 1, we argue why the result is true when rewards do not satisfy all-honest-profitability (Definition 6.2). For all-honest-profitability, the proof is further divided in Steps 2,3. In Step 2, we find an environment Z_2 for any adversary A_2 with a slight advantage in the probability of mining (such as due to selfish mining). In Step 3, we complete the proof by showing A_2 has a higher expected payoff in environment Z_2 than any front-running semi-honest adversary $A_1 \in A_{fr}$. The complete proof is provided in Appendix C.2. □

Theorem 6.4. *For attacks A_2 which extend for less than α_{th} phases, PRAGTHOS with deflationary (ϑ is geometrically decreasing) block-reward scheme is strongly attack-payoff secure against a computationally bounded adversary $A \in \mathcal{A}^{\alpha_{th}}$ for $\beta_{adv} < \frac{1}{2}$ where,*

$$\alpha_{th} = 1 + \lfloor \frac{log(1 - p_{fr})}{log(\vartheta)} \rfloor$$

Here p_{fr} is the probability that the protocol accepts a query by a front-running semi-honest adversary.

Proof. Proof of this theorem uses the adversary discussed in Theorem 6.3. This adversary is the smallest powerful adversary that can achieve a greater payoff from any front-running strategy. We bound the attacker to be weaker than this adversary to obtain the result. The complete proof is given in Appendix C.3 □

7 Conclusion and Future Work

Conclusion. In this paper, we analyzed and found security attacks possible on blockchain protocols. E.g., Bitcoin is not secure against adversary control $.45$ fraction of the computing power. We identified reasons why previous security analysis models fail to capture these. Towards this, we proposed a novel model of Rational Protocol Design, PRPD. Using this, we designed solutions to address these attacks and proposed a framework for designing PoW blockchain protocols, namely, **PRAGTHOS**. We proved that **PRAGTHOS** is strongly attack-payoff secure under an inflationary block-reward scheme. Under a deflationary block-reward scheme, we prove that **PRAGTHOS** is secure against an adversary bounded by the number of rounds.

Future Work. The model used for security analysis of PoW blockchain protocol fails to capture rational deviations which are incentivized from outside the system, such as the attacks proposed in Judmayer et al. [2020]. We believe such attacks can be captured through the generalization of PRPD. We believe our results expand the existing models of Game-Theoretic security of Blockchains to a more general model. Extension of models of security for other types of blockchain protocols, such as PoS and other cryptographic protocols against incentive-driven adversaries, might be of interest and is left for future work.

References

Satoshi Nakamoto. Bitcoin : A peer-to-peer electronic cash system, 2009.

Vitalik Buterin et al. Ethereum white paper, 2013.

Nick Szabo. Formalizing and securing relationships on public networks. *First Monday*, 2, 1997.

Michael J. Fischer, Nancy A. Lynch, and Mike Paterson. Impossibility of distributed consensus with one faulty process. In *Principles of Database Systems (PODS) '83*, 1983.

Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol: Analysis and applications. In *EUROCRYPT*, 2015.

Juan A. Garay, Aggelos Kiayias, and Nikos Leonardos. The bitcoin backbone protocol with chains of variable difficulty. In *CRYPTO*, 2017.

Ittay Eyal and Emin Gün Sirer. Majority is not enough: Bitcoin mining is vulnerable. In *Financial Cryptography*, 2014.

Christian Badertscher, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. Bitcoin as a transaction ledger: A composable treatment. *IACR Cryptol. ePrint Arch.*, 2017:149, 2017.

Ran Canetti. Universally composable security: a new paradigm for cryptographic protocols. *Proceedings 2001 IEEE International Conference on Cluster Computing*, pages 136–145, 2001.

Juan A. Garay, Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. Rational protocol design: Cryptography against incentive-driven adversaries. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 648–657, 2013.

Christian Badertscher, Juan A. Garay, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. But why does it work? a rational protocol design treatment of bitcoin. *IACR Cryptol. ePrint Arch.*, 2018:138, 2018.

Christian Badertscher, Yun Lu, and Vassilis Zikas. A rational protocol treatment of 51% attacks. In *IACR Cryptol. ePrint Arch.*, 2021.

Aljosha Judmayer, Nicholas Stifter, Alexei Zamyatin, Itay Tsabary, Ittay Eyal, Peter Gazi, Sarah Meiklejohn, and Edgar R. Weippl. Sok: Algorithmic incentive manipulation attacks on permissionless pow cryptocurrencies. In *IACR Cryptol. ePrint Arch.*, 2020.

Kevin Liao and Jonathan Katz. Incentivizing double-spend collusion in bitcoin. 2017a.

Kevin Liao and Jonathan Katz. Incentivizing blockchain forks via whale transactions. In *Financial Cryptography Workshops*, 2017b.

Runchao Han, Zhimei Sui, Jiangshan Yu, Joseph K. Liu, and Shiping Chen. Fact and fiction: Challenging the honest majority assumption of permissionless blockchains. *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, 2021.

Itay Tsabary and Ittay Eyal. The gap game. *Proceedings of the 11th ACM International Systems and Storage Conference*, 2018.

Shoeb Siddiqui, Ganesh Vanahalli, and Sujit Gujar. Bitcoinf: Achieving fairness for bitcoin in transaction-fee-only model. In *Autonomous Agents and Multiagent Systems, (AAMAS)*, 2020.

Lin Chen, Lei Xu, Zhimin Gao, Ahmed Imtiaz Sunny, Keshav Kasichainula, and W. Shi. A game theoretical analysis of non-linear blockchain system. In *Autonomous Agents and Multiagent Systems, AAMAS*, 2021.

Dimitris Karakostas, Aggelos Kiayias, and Thomas Zacharias. Blockchain nash dynamics and the pursuit of compliance. *ArXiv*, abs/2201.00858, 2022.

Sarah Azouvi and Alexander Hicks. Sok: Tools for game theoretic models of security for cryptocurrencies. *ArXiv*, abs/1905.08595, 2020.

Puwen Wei, Quan Yuan, and Yuliang Zheng. Security of the blockchain against long delay attack. *IACR Cryptol. ePrint Arch.*, 2018:800, 2018.

Peter Gazi, Ling Ren, and Alexander Russell. Practical settlement bounds for proof-of-work blockchains. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, CCS '22, page 1217–1230, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450394505. doi:10.1145/3548606.3559368. URL `https://doi.org/10.1145/3548606.3559368`.

Quan Yuan, Puwen Wei, Keting Jia, and Haiyang Xue. Analysis of blockchain protocol against static adversarial miners corrupted by long delay attackers. *Science China Information Sciences*, 63:1–15, 2020.

Peyman Momeni, Sergey Gorbunov, and Bohan Zhang. Fairblock: Preventing blockchain front-running with minimal overheads. Cryptology ePrint Archive, Paper 2022/1066, 2022.

Mike Graf, Daniel Rausch, Viktoria Ronge, Christoph Egger, Ralf Küsters, and Dominique Schröder. A security framework for distributed ledgers. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 1043–1064, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384544. doi:10.1145/3460120.3485362. URL `https://doi.org/10.1145/3460120.3485362`.

Junming Ke, Pawel Szalachowski, Jianying Zhou, and Qiuliang Xu. Formalizing bitcoin crashes with universally composable security. In *IACR Cryptol. ePrint Arch.*, 2020.

Malte Möser and Rainer Böhme. Trends, tips, tolls: A longitudinal study of bitcoin transaction fees. In Michael Brenner, Nicolas Christin, Benjamin Johnson, and Kurt Rohloff, editors, *Financial Cryptography and Data Security*, pages 19–33, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg. ISBN 978-3-662-48051-9.

Juanjuan Li, Yong Yuan, Shuai Wang, and Fei-Yue Wang. Transaction queuing game in bitcoin blockchain. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 114–119, 2018. doi:10.1109/IVS.2018.8500403.

Nicolas Houy. The economics of bitcoin transaction fees. *ERN: Other Microeconomics: Production*, 2014.

Tim Roughgarden. Transaction fee mechanism design. *ACM SIGecom Exchanges*, 19:52 – 55, 2021.

Miles Carlsten, Harry Kalodner, S. Matthew Weinberg, and Arvind Narayanan. On the instability of bitcoin without the block reward. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, CCS '16, page 154–167, New York, NY, USA, 2016. Association for Computing Machinery. ISBN 9781450341394. doi:10.1145/2976749.2978408. URL `https://doi.org/10.1145/2976749.2978408`.

Cynthia Dwork, Nancy A. Lynch, and Larry J. Stockmeyer. Consensus in the presence of partial synchrony (preliminary version). In *ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing*, 1984.

Shayan Eskandari, Seyedehmahsa Moosavi, and Jeremy Clark. Sok: Transparent dishonesty: Front-running attacks on blockchain. *Econometrics: Computer Programs & Software eJournal*, 2019.

Lorenz Breidenbach, Philip Daian, Florian Tramèr, and Ari Juels. Enter the hydra: Towards principled bug bounties and exploit-resistant smart contracts. In *Proceedings of the 27th USENIX Conference on Security Symposium*, SEC'18, page 1335–1352, USA, 2018. USENIX Association. ISBN 9781931971461.

Harry A. Kalodner, Miles Carlsten, Paul Ellenbogen, Joseph Bonneau, and Arvind Narayanan. An empirical study of namecoin and lessons for decentralized namespace design. In *Workshop on the Economics of Information Security*, 2015.

Mark Friedenbach. Forward blocks on-chain / settlement capacity increases without the hard-fork. 2018.

Gregory W. Shirah. The verge. In *International Conference on Computer Graphics and Interactive Techniques*, 2008.

Sheldon M. Ross. Introduction to probability models. *Technometrics*, 40:78–78, 1975.

En.bitcoin.it. Bitcoin script - bitcoin wiki, 2021. URL `Available:https://en.bitcoin.it/wiki/Script`.

En.bitcoin.it. Pay to script hash - bitcoin wiki, 2017. URL `https://en.bitcoin.it/wiki/Pay_to_script_hash`.

Ahto Buldas, Risto Laanoja, and Ahto Truu. Keyless signature infrastructure and pki: hash-tree signatures in pre- and post-quantum world. *Int. J. Serv. Technol. Manag.*, 23:117–130, 2017.

Savva Shanaev, Arina Shuraeva, Mikhail Vasenin, and Maksim Kuznetsov. Cryptocurrency value and 51% attacks: Evidence from event studies. In *The Journal of Alternative Investments*, 2019.

Michael Mitzenmacher and Eli Upfal. *Probability and Computing: Events and Probability*. 2005.

Joseph Bonneau. Hostile blockchain takeovers (short paper). In *Financial Cryptography Workshops*, 2018.

A Proofs of Theorems Regarding Attacks

A.1 Proof of Theorem 3.1

Step 1: Note that the utility of the adversary is

$$U_A = \max_{\mathcal{Z} \in ITM} \min_{\mathcal{S} \in C_A} E[v_A^{\mathcal{G}_{weak-ledger}, \mathcal{S}, \mathcal{Z}}]$$

Since the objective of Protocol Descriptoris to ensure security in the worst-case, we consider an environment that maximizes U_A. The adversary optimally chooses the parameters (α, r_1, r_2) under its control as follows:

❶ to maximize the probability of successful attack, the adversary maximizes the duration, i.e., it sets $r_1 = 0, r_2 = 1$.

❷ For the adversary, to launch attack, it is a best strategy is to adjust α such that $\tau_{adv} = \tau_{min}$ which is achieved when it sets $\alpha = \frac{\beta_{adv}}{\tau_{min}}$.

With these parameters, Equation 3 reduces to
$\frac{\tau_{hon}}{1-\beta_{adv}} + \frac{1}{1-\beta_{adv}} > \frac{1}{\beta_{adv}} + \frac{\tau_{adv}}{\beta_{adv}} \Rightarrow \frac{\beta_{adv}}{1-\beta_{adv}} > \frac{1+\tau_{min}}{1+\tau_{hon}}$
Step 2: Our analysis only concerns $\beta_{adv} < \frac{1}{2}$ because for $\beta_{adv} > \frac{1}{2}$ forking using the 51% attack Shanaev et al. [2019] is always possible. Additionaly, the attack being a security attack, EO observes the attack and thus, by definition RP follows honest strategy and we treat them as honest. We therefore show the result for $\tau_{min} < \frac{1}{2} \Rightarrow 1-\beta_{adv} > \frac{1}{2} > \tau_{min}$.
Step 3: With the parameters set as described in Steps 1 and 2, we have the following inequality:

$$\frac{2\beta_{adv} - \beta_{adv}^2 - 1 + \beta_{adv}}{1 - \beta_{adv}} > \tau_{min}$$

On solving for β_{adv}, we have

$$\beta_{adv}^2 - (3 + \tau_{min})\beta_{adv} + (1 + \tau_{min}) < 0$$

The roots of the equation are L_1, L_2, where

$$L_1 = \frac{(3 + \tau_{min}) - \sqrt{(3 + \tau_{min})^2 - 4(\tau_{min} + 1)}}{2}$$

$$L_2 = \frac{(3 + \tau_{min}) + \sqrt{(3 + \tau_{min})^2 - 4(\tau_{min} + 1)}}{2}$$

The feasible region for β_{adv} is (L_1, L_2). However, $L_2 > 1$, so the intersection of possible β_{adv} values with values feasible for DIFFICULTY ALTERING attack gives us the bound $\beta_{adv} \geq \frac{(3+\tau_{min})-\sqrt{(3+\tau_{min})^2-4(\tau_{min}+1)}}{2}$.
Step 4: Let Q_i^{adv} and Q_i^{hon} be random variables (RV) denoting the time taken to mine one block by the adversary and the HP respectively. We define two RVs Q^{adv} and Q^{hon} as follows. $Q^{adv} = \sum_{i=0}^{2*\lambda} Q_i^{adv}$ and $Q^{hon} = \sum_{i=0}^{2\lambda} Q_i^{hon}$. Q^{adv} and Q^{hon} denote the total time to mine $2 * \lambda,$ blocks;. The factor 2 is because attack progresses for 2 phases (since $1 - r_1 = r_2 = 1$). Since RVs Q_i^{adv}s (similarly Q_i^{hon}) are independent of each other for different values of i, we can apply Chernoff bound (from Mitzenmacher and Upfal [2005] Equation 4.2 and Equation 4.5)

$$Pr[Q^{adv} \geq (1 + \varepsilon)E[Q^{adv}]] < e^{-\frac{E[Q^{adv}]\varepsilon^2}{3}}$$

$$Pr[Q^{adv} \leq (1 - \varepsilon)E[Q^{adv}]] < e^{-\frac{E[Q^{adv}]\varepsilon^2}{2}}$$

Summing up the deviation probabilities, the expected time to mine $\lambda(r_1 + r_2)$ blocks deviates by more than ε with probability $negl(\Theta\varepsilon^2)$, as Expected time to mine a block is $\propto \Theta$. Therefore, the DIFFICULTY ALTERING Attack is successful with probability $> 1 - 2 \cdot negl(\Theta\varepsilon^2)$.

A.2 Proof of Thm. 3.2

We prove this in four steps by calculating payoffs on deviating and following the protocol (Step 1 & 2). Then comparing them to derive the bound (Step 3) and finally calculating the probability of success of this attack (Step 4).

Step 1 **Payoff on deviating:** On deviation, the last k blocks of P_i of chain C_H are orphaned, and the mining cost spent by the AP, RP and HP on the main chain is wasted for these blocks. This reduces $k\beta_{par}\chi_1$ from the payoff for $par \in \{rat, adv\}$. In addition, Mining on the C_A incurs cost = $\frac{(k+m)\beta_{par}\chi_1}{\beta_{rat}+\beta_{adv}}$. This is higher as HP are not mining on

C_A. In addition, the block reward collected by par is $\frac{\beta_{par}}{\beta_{rat}+\beta_{adv}}$ of the total block reward of these $k+m$ blocks in the main chain. This value is equal to $\frac{(k+m\vartheta)r_{block}\beta_{par}}{\beta_{adv}+\beta_{rat}}$. Combining all three, we get the expected payoff on deviating as:

$$v_{par}(\pi^{'}) = \frac{n-1}{n}\frac{\beta_{par}}{\beta_{rat}+\beta_{adv}}\big(r_{block}(k+m\vartheta)-\chi_1(k+m)\big)-k\beta_{par}\chi_1$$

QUICK FORK Attack is successful w.p. $\frac{n-1}{n}$ which we show in Step 4 of the proof.

Step 2 **Payoff on following:** On following the protocol, the payoff for par (RP or AP) is $r_{block}(k+m\vartheta)\beta_{par}$ and the cost incurred is $\chi_1(k+m)\beta_{par}$. Therefore, the total payoff is:

$$v_{par}(\pi) = \beta_{par}\big(r_{block}(k+m\vartheta)-(k+m)\chi_1\big)$$

We can rewrite $v_{par}(\pi^{'})$ as $v_{par}(\pi^{'}) = \frac{n-1}{n}(\frac{v_{par}(\pi)}{\beta_{rat}+\beta_{adv}} - \chi_1 k\beta_{par})$. Further, we substitute $\eta = \frac{r_{block}}{\chi_1}$

Step 3: The condition $v_{par}(\pi^{'}) - v_{par}(\pi) > 0$

$$\Rightarrow \chi_1\beta_{par}(\frac{(n\beta_{hon}-1)}{n(1-\beta_{hon})})((\eta-1)k+(\eta\vartheta-1)m > \chi_1\beta_{par}k$$

$$\Rightarrow (\frac{(n\beta_{hon}-1)}{n(1-\beta_{hon})})((\eta-1)k+(\eta\vartheta-1)m > k$$

Since C_A overtakes C_H by m blocks of phase P_{i+1}, therefore, the time taken to mine m blocks in C_H by β_{hon} mining power is $\geq$ as the time taken to mine $k+m$ blocks in C_A by $\beta_{rat}+\beta_{adv}$ mining power. For notational ease, we represent $J = \frac{\beta_{hon}}{1-2\beta_{hon}}$. Thus,

$$\frac{m}{\beta_{hon}} \geq \frac{m+k}{1-\beta_{hon}} \Rightarrow m \geq \frac{k\beta_{hon}}{1-2\beta_{hon}} = kJ$$

Clearly, as $m>0$, $\beta_{hon} < \frac{1}{2}$,implying $\beta_{adv}+\beta_{rat} > \frac{1}{2}$. We take the earliest possible m, which gives us $m = \frac{k\beta_{hon}}{1-2\beta_{hon}}$. This gives us $v_{par}(\pi^{'}) - v_{par}(\pi) > 0$ as

$$\Rightarrow (\frac{(n\beta_{hon}-1)}{n(1-\beta_{hon})})(\eta-1+J\eta\vartheta-J) > 1$$

after substituting J and simplifying, we get

$$\eta > \frac{1-\beta_{hon}}{n\beta_{hon}-1}\cdot\frac{n-\beta_{hon}n-1}{1-(2-\vartheta)\beta_{hon}}$$

Step 4: For the attack to be successful, the lead of HP should drop from k to 0 before it reaches ρ. This can be solved as Gambler's ruin problem (Sec 4.5.1 Ross [1975]) with random walk moving in favor of HP with probability β_{hon}. With this, the probability of an attack is $\frac{1-\phi^{\rho-k}}{1-\phi^{\rho}}$ where $\phi = \frac{\beta_{hon}}{\beta_{rat}+\beta_{adv}}$. This probability is greater than $\frac{n-1}{n}$ if. $\frac{1+(n-1)\phi^{\rho}}{n} > \phi^{\rho-k}$. We can simplify this as

$$k < \lfloor \rho - log_{\phi}(\frac{1+(n-1)\phi^{\rho}}{n})\rfloor \tag{8}$$

With this, k the probability of the QUICK FORK attack being successful is at least $> \frac{n-1}{n}$.

A.3 Proof of Thm. 3.3

Proof. For proof of this attack, we consider that the bribe amount is a z fraction of the Block reward for a single block ($z>0$, but a small value). The behavior of AP, RPand HP follows as described in the Attack-strategy in Section 3.2.2. In case of a tie between AP and HP blocks, if AP's mined block becomes part of the longest chain, the payoff is $(1-z)r_{block}$ for the adversary, and the party which mines block on top of AP block gets payoff $(1+z)r_{block}$.

Bribes incentivize RPs to deviate from the protocol. Due to this, it is safe to consider the same payoffs as in Eyal and Sirer [2014]; however, the γ – the fraction of non-adversarial parties mining on the adversarial block in the case of a tie for the longest chain is $\frac{\frac{\beta_h}{2}+\beta_r}{\beta_h+\beta_r}$. Using this value in the result from Eyal and Sirer [2014], also given in Equation 2, we get $\frac{1}{2} > \beta_a > \frac{\beta_h}{2\beta_r+4\beta_h}$ □

A.4 Proof for Lemma 3.4

Proof. First, we calculate the utility for a rational party i for following the protocol Π_{Gossip}, i.e., broadcasting a transaction $\mathfrak{T}$ with transaction fee tx_i that it hears. Then, we calculate its utility for not broadcasting $\mathfrak{T}$. We account for the unfavourable events (unfavourable for the attacker) that (i) some other parties may add $\mathfrak{T}$ and (ii) discounting the rewards if $\mathfrak{T}$ is added by the party later. We then argue that later leads to a higher utility.

Broadcasting $\mathfrak{T}$ The probability of a single party mining a block in round t is $p_{su}(t)$, and they can make q queries in each round. Then, the utility in following the gossip protocol is given for a party with β_i fraction of mining power in their control as :

$$u_i^*(\beta_i) = (\sum_{j=1}^{n} tx_j)(\sum_{t=1}^{\infty} \delta^t p_{su}(t)(1-p_{su}(t))^{n(t)\cdot t})$$

Not Broadcasting $\mathfrak{T}$ The utility for party i becomes,

$$u_i^{'}(\beta_i) = (\sum_{j=1, j\neq i}^{n} tx_j)(\sum_{t=1}^{\infty} \delta^t p_{su}(t)(1-p_{su}(t))^{n(t)\cdot t}) + \frac{tx_i}{p_{su}}$$

As $\mathfrak{T}$ is accepted with probability p_{su} in each round. Consider K is the random variable 1 when the party mines a block and 0 otherwise. Then, K is a geometric random variable with $E[K] = \frac{1}{p_{su}}$. We can clearly see that $u_i^*(\beta_i) < u_i^{'}(\beta_i)$. Therefore σ_{tw} dominates Π_{gossip}. □

B Proofs of Detering Attacks

B.1 Proof for Lemma 5.3

Proof. Our proof proceeds in three steps. In Step 1, we define an environment (Z_1, S_1) for $S_1 \in \mathcal{A}_{fr}$. In Step 2, we define (Z_2, S_2) where $S_2 \in \mathcal{A}_{SMB}$. We then show in Step 3 that both these executions are indistinguishable from each other.

Step 1: Consider a simulator $S_1 \in \mathcal{A}_{fr}$ be a semi-honest adversary. Environment Z_1 is such that it observes all chains, and if there is a contest between two chains that are at the same height, they maximally delay messages from miners mining on the chain with the last block not mined by party corrupted by S_1.

Step 2: Consider any simulator simulating SELFISH MINING WITH BRIBING attack ($S_2 \in \mathcal{A}_{SMB}$) and any general environment Z_2, which communicates messages in the same (partially-synchronous) manner for both AP, RP and HP.

Step 3: It is clear by comparison that for any party viewing the two systems, $E[v^{\Pi, S_1, Z_1}] \equiv E[v^{\Pi, S_2, Z_2}]$, where v represents the external view of the system. This means that for any PPT ITM it is not possible to distinguish between the two systems. Let D be the discriminator which is a PPT ITM. Let us denote two systems $\rho_1 =< \Pi, S_1, Z_1 >$ and $\rho_2 =< \Pi, S_2, Z_2 >$. We denote a random variable C which can take value 1 and 2 with equal probability. We select a system $\rho = \rho_C$ to be sent to the discriminator D based on the value of C. Then, if D is a PPT ITM, then $Pr[D(\rho) = C] = \frac{1}{2} + negl(\varepsilon)$.

Thus, because the two environments are indistinguishable we cannot ensure attack-payoff security (adversary follows $A \in \mathcal{A}_{fr}$ without also allowing SELFISH MINING WITH BRIBING.

□

B.2 Proof of Lemma 5.4

Let $Pool$ be the group of parties collectively deviating from $\Pi_{tx-inclusion}$ and withhold a set of transactions with a cumulative transaction fee tx. Let β_{Pool} be their collective computing power.

First, we focus on and compare the utility $Pool$ obtains from following gossip protocol versus TRANSACTION WITHHOLDING. Next, since the analysis is on the transaction network, we assume the PoW protocol to be followed correctly. For the sake of abstraction, we consider the probability of mining a block in a round t by a single party is $p_{su}(t)$ and the total number of parties are $n(t)$. The payoff on following the gossip protocol becomes :

$$v(\Pi_{gossip}) = \sum_{i=1}^{\infty} \delta^{i-1} v_i(\Pi_{gossip})$$

```
$\mathcal{Z}_1^{Network}$

on RECEIVE()
  $M \leftarrow \{0,1\}^*$ (receive broadcasted message)
  if sender($M$) is $\mathcal{A}$
    - $B_{\mathcal{A}} \leftarrow B_{\mathcal{A}} \cup \ M$
  else
    - broadcast first $max(2, |B_{\mathcal{A}}|)$ messages of $B_{\mathcal{A}}$
    - broadcast $M$
```

Figure 4: Environment Z_1

where δ is the discount factor, which captures the increasing chance of the proposer sending the transaction to another party, thereby reducing the chance of the current party exclusively mining for that transaction. v_i is the expected payoff from the transaction in the i^{th} round.

$v_i(\Pi_{gossip}) =$

$$(1-(1-p_{su}(i))^{\beta_{Pool}n(i)})(1-p_{su}(i))^{n(i)\cdot(i-1)}\frac{tx}{2^l}$$

On summing up $\sum_{i=0}^{\infty} v_i(\pi_{gossip})$, we get

$$\sum_{i=0}^{\infty}\delta^i(1-(1-p_{su}(i+1))^{\beta_{Pool}n})(1-p_{su}(i+1))^{n(i+1)\cdot i}\frac{tx}{2^l}$$

For the deviating protocol (Π_{secret}), the utility in the i^{th} round $v_i(\Pi_{secret})$ becomes,

$$(1-(1-p_{su}(i))^{\beta_{Pool}n(i)})(1-p_{su}(i))^{n(i)\cdot(i-1)\beta_{Pool}}\frac{tx}{2^l}$$

We therefore have

$$v_i(\Pi_{secret}) - v_i(\Pi_{gossip}) = (1-(1-p_{su}(i))^{\beta_{Pool}n(i)})\frac{tx}{2^l}$$
$$(1-p_{su}(t))^{n(i)\cdot(i-1)\beta_{Pool}}(1-(1-p_{su}(t))^{n(i)\cdot(i-1)(1-\beta_{Pool})})$$

If we sum this over geometrically decreasing δ, we get

$$v(\Pi_{secret}) - v(\Pi_{gossip}) = \sum_{i=1}^{\infty}(1-(1-p_{su}(i))^{\beta_{Pool}n(i)})\frac{tx}{2^l}\mathfrak{R}$$
$$\mathfrak{R} = (1-p_{su}(i))^{n(i)\cdot(i-1)\beta_{Pool}}(1-(1-p_{su}(i))^{n(i)\cdot i(1-\beta_{Pool})})$$

Taking upper limit of probability as 1 for $(1-(1-p_{su}(i))^{n(i)\cdot i(1-\beta_{Pool})})$, we get

$$\mathfrak{R} \leq (1-p_{su}(i))^{n(i)\cdot(i-1)\beta_{Pool}} = \mathfrak{M}$$

Therefore $\mathfrak{M}$ is the probability that no miner mines a block at round i.

$$v(\Pi_{secret}) - v(\Pi_{gossip}) \leq \sum_{i=1}^{\infty}(1-(1-p_{su}(i))^{\beta_{Pool}n(i)})\frac{tx}{2^l}\mathfrak{M}$$

If we upper bound the remaining probability term to 1, we get the expression

$$v(\Pi_{secret}) - v(\Pi_{gossip}) \leq \frac{tx}{2^l}\sum_{i=0}^{\infty}\mathfrak{M}$$

Since total probability of the block not being mined ($\sum_{i=1}^{\infty}\mathfrak{M} \leq 1$) we get the bound

$$v(\Pi_{secret}) - v(\Pi_{gossip}) \leq \frac{tx}{2^l}$$

Since tx is polynomial in l, $\frac{tx}{2^l}$ is negligibly small. Let $\epsilon_G = O(tx \cdot 2^{-l})$. We therefore get $v(\Pi_{secret}) - v(\Pi_{gossip}) \leq \epsilon_G$

C Proofs for PRAGTHOS Analysis

The proofs for 6.2 and 6.3 are in the Attack model $< \mathcal{F}, < \mathcal{F} >, \overline{v} >$ where as decribed in 4.2, the attack model is $< \mathcal{G}_{ledger}, \mathcal{G}_{weak-ledger},$
$(v_A, v_R, v_H) >$

C.1 Proof for Lemma 6.2

Proof. Step 1a Let $A_1 \in A_{fr}$ be a front-running adversary which makes q^* queries. Further consider the environment Z_1 which runs the execution of the protocol where the adversary is activated to make q^* queries before halting, and the HPs are activated till at least one of them output all the blocks mined by the adversary in their longest chain. We consider real-world UC execution, and all random variables are correspondingly defined.

Consider random variable X_i which is 1 if i^{th} query by adversary successfully mines a block, and 0 otherwise. Thus, the payoff for APin q^* queries is

$$\mathcal{R}_{A_1} = \sum_{i=1}^{q^*} X_i r_{block} \theta - \chi$$

Notice that we exclude the payoff that the adversary gets by decreasing the value of coin by lowering $\theta(t)$ through security attacks – forking the chain, because in **PRAGTHOS**, forking is not possible for $\beta_{adv} < \frac{1}{2}$.

Step 1b Since due to variation in the number of miners, the difficulty of mining, and hence the probability of a block getting accepted changes with each epoch, the probability of getting a block accepted is p_e in epoch e. As the number of miners do not grow exponential across rounds, we can assume that there exists a polynomial $s(t)$, such that $n(t) \leq s(t)\ \forall t$. Let p_e^{min} be the probability of A_1 mining a block in round t if $n(t)$ grows exactly as $s(t)$. Clearly, $\forall$ epochs, $p_e \geq p_e^{min}$. Let $p_{min} = \min_{\forall e} p_e^{min}$.

Step 1c Consider that the sum of the block-reward up to a queries is denoted by $\mathfrak{J}(a) \cdot r_{block}(0)$.

$$\mathfrak{J}(a) := \sum_{i=0}^{a} \frac{r_{block}(i)}{r_{block}(0)} \approx \sum_{j=0}^{\alpha_a} \Lambda \vartheta(j)$$

We have replaced queries with number of blocks mined, because the expected number of blocks mined deviates very less for large number of queries, and in q queries, the number of blocks mined does not deviate by more than negligible amount with overwhelming probability. This result follows from the Chernoff-bound analysis. Since both θ and χ are variable for rounds 1 to r_1 till which the protocol runs, we define $\eta_{max} = \max_{t \in [1, r_1]} \frac{\chi}{p_e \theta(t) r_{block}}$. Since mining is profitable, we can conclude that $\eta_{max} < 1$. Also $\theta = \theta(t)$ is same $\forall\ t$ because both Π and Adversarial strategy is fixed. Now, we can conclude that

$$E[\mathcal{R}_{A_1}] \geq \mathfrak{J}(q^*)(1 - \eta_{max}) p_{min} r_{block} \theta$$

Step 2 Now, consider any arbitrary adversary $A_2 \in ITM$. This adversary makes Q queries during its execution in an environment Z. Let P_Q be the distribution of the number of queries made by this adversary $q = \max \text{support}(P_Q)$. Let Z_2 be environment where $Q = q$. Consider this (A_2, Z_2), where the expected payoff of the adversary is upper bounded by taking probability of mining per query as 1, which gives us

$$E[\mathcal{R}_{A_2}] \leq \mathfrak{J}(q) r_{block} \theta$$

This upper bound comes from the fact that a single query can extend the blockchain by at most one block, in the $\mathcal{F}$ functionality. However, if that is not the case with it's UC-Realization, the state exchange protocol Π, then $EXEC_{\mathcal{F}, \mathcal{S}_2, \mathcal{Z}_2} \not\approx EXEC_{\Pi_B, \mathcal{A}_2, \mathcal{Z}_2}$, and thus $C_{A_2} = \phi$, which is not possible.

Step 3a We choose Z_1 such that the condition $\mathfrak{J}(q^*) \geq \frac{\mathfrak{J}(q)\kappa}{(1-\delta)(1-\eta_{max})p_{min}}$ is satisfied. Because the crypto-currency is inflationary in nature, we are assured that such an environment always exists. This is because for inflationary series $\mathfrak{J}$, there always exist $a_2 > a_1$ such that $\mathfrak{J}(a_2) > \mathfrak{J}(a_1)$,

$$\because \lim_{a_2 \to \infty} \mathfrak{J}(a_2) - \mathfrak{J}(a_1) \longrightarrow \infty\ \forall\ a_1$$

Now, consider the probability that the payoff of A_1 is less than the expected payoff of A_2. We can say there always exist such Z_1 for each S_2, because there always exist such q^* for q, due to the diverging nature of the series ϑ.

$$Pr[\mathcal{R}_{A_1} < E[\mathcal{R}_{A_2}]] \leq Pr[\mathcal{R}_{A_1} < \mathfrak{J}(q) r_{block} \theta]$$

$$\leq Pr[\mathcal{R}_{A_1} < (1-\delta)(1-\eta_{max})\mathfrak{I}(q^*)r_{block}p_{min}\theta]$$
$$\leq Pr[\mathcal{R}_{A_1} < (1-\delta)E[\mathcal{R}_{A_1}]] = Pr[\sum_{i=1}^{q^*} X_i \leq (1-\delta)E[\sum_{i=1}^{q} X_i]$$
$$< e^{-\frac{\delta^2 q^* p}{2}}$$

The last inequality follows from Chernoff's bound Mitzenmacher and Upfal [2005]. The difference in expected values of all Random variables $(Q, \mathcal{R}_{A_1}, \mathcal{R}_{A_2}, X_i ...)$ in both Real and Ideal execution cannot be more then a small amount ϵ. This is because, if the expected value deviates by more than ϵ, this event is observable, and by Chernoff-bound analysis[19] it's value is very small. Therefore, such an event happening means that $EXEC_{\Pi_B,A,Z} \not\approx EXEC_{\mathcal{F},S,Z}$ with very high probability; which contradicts that $S \in C_A$. Therefore, for ideal payoff v_A, we have $E[v_A^{\mathcal{F},S_1,Z_1}] \approx E[\mathcal{R}_{A_1}]$.

Using this, we can conclude the result that $E[v_A^{\mathcal{F},S_1,Z_1}] \geq S(q)r_{block}\theta \geq E[v_A^{\mathcal{F},S_2,Z_2}]$ with overwhelming probability.

Step 3b Now, we need to show $U_A(\Pi, \mathcal{A}_1) + negl(\kappa) \geq U_A(\Pi, \mathcal{A}_2)$. Let us assume this is false. That means $\exists S_2, Z_2$ such that

$$\max_{Z_2 \in ITM}\left\{\min_{S_2 \in C_{A_2}}\left\{E[v_A^{\mathcal{F},S_2,Z_2}]\right\}\right\} \geq$$
$$\max_{Z_1 \in ITM}\left\{\min_{S_1 \in C_{A_1}}\left\{E[v_A^{\mathcal{F},S_1,Z_1}]\right\}\right\}$$

This means that there exists a Z_2 such that $\forall Z_1$

$$\min_{S_2 \in C_{A_2}} E[v_A^{\mathcal{F},S_2,Z_2}] \geq \min_{S_1 \in C_{A_1}} E[v_A^{\mathcal{F},S_1,Z_1}]$$

But we have shown that this is false, except with negligible probability. This means that our assumption was wrong, and in reality, $\forall A_2 \exists A_1$ such that the attack-payoff security condition holds. □

C.2 Proof for Theorem 6.3

Proof. It should be noted that we discuss this scheme for geometrically decreasing converging series, because this is the series that is employed in most of the existing PoW blockchains. But, the result holds for any converging $< \vartheta >$.

Step 1 (No all-honest-profitability): Since the protocol initially (when started) follows all honest profitability, for this case, the condition $r_{block}(t)p_{hon}(t)\theta(t) < \chi$ is becomes true after some rounds. The dominant strategy for HP and RP is to abstain from the protocol. However, for AP, this scenario could still be profitable by shorting the cryptocurrency (following $\mathcal{A}_{\overline{ahp}}$).

$\mathcal{A}_{\overline{ahp}}$

1. Hold *short position* against the cryptocurrency in some round when ahp[a] is satisfied.
2. When $\overline{ahp}$, $\beta_{adv} = 1$, because all RP and HP leave the protocol.
3. When $\beta_{adv} > \frac{1}{2}$, launch security attack against the protocol bringing the value of the crypto-currency down and profit from the short position held.

[a] all-honest-profitability

Step 2 (All-honest-profitability): For protocol to be not attack-pay-off-secure, we need to show the existence of an attack strategy A_2, which is simulated by simulator S_2 in environment Z_2, such that $\nexists A_1 \in A_{fr}$ for any environment Z_1 for which the following equation holds

$$E[v_A^{\mathcal{F},S_1,Z_1}] < E[v_A^{\mathcal{F},S_2,Z_2}]$$

Here, all, $A_1, A_2, S_1, S_2, Z_1 \& Z_2$ are PPT ITMs[20] .Let us consider any attack where the probability of acceptance of the block by the adversary per query is p_{attack}, which is greater than the same probability for a front-running semi-honest

[19] $Pr[X_{ideal} \geq (1-\epsilon)E[X_{real}]] < exp(-\frac{\epsilon^2 E[X_{real}]}{2})$

[20] probabilistic polynomial time interactive Turing machines

miner p_{fr}. Note that at least SELFISH MINING WITH BRIBING, $p_{attack} > p_{fr}$. Our proof being general, we consider any attack where this is true. Now, for the attack-strategy A_2, which is simulated by simulator $S_2 \in C_{A_2}$, and makes q queries, which are distributed across α_{opt} phases. Consider the expected payoff for this adversary

$$E[\mathcal{R}_{A_2}] = \Lambda\theta r_{block} p_{attack} \sum_{i=0}^{\alpha_2 - 1} \vartheta^i - q\chi$$

$$= \Lambda\theta r_{block} p_{attack} \frac{1 - \vartheta^{\alpha_2}}{1 - \vartheta} - q\chi$$

Now consider any front-running adversary $A_1 \in A_{fr}$, which is simulated by $S_1 \in C_{A_1}$ and the environment Z_1 which allows the adversary to make q^* queries and waits for at least HP to output the chain with all adversarial blocks in it, before halting the execution. Consider that in expectation $\Lambda\alpha_1$ blocks are mined in this duration. The expected payoff of such an adversary is $E[\mathcal{R}_{A_1}] = \Lambda\theta r_{block} p_{fr} \sum_{i=0}^{\alpha_1 - 1} \vartheta^i - q^*\chi$, i.e.,

$$E[\mathcal{R}_{A_1}] = \Lambda\theta r_{block} p_{fr} (\frac{1 - \vartheta^{\alpha_2}}{1 - \vartheta} + \vartheta^{\alpha_2} \sum_{j=0}^{\alpha_1 \vartheta^j - \alpha_2}) - q^*\chi$$

Since $p_{attack} > p_{fr}$, let $p_{attack} - p_{fr} = \Delta$. We want α_{opt} as the optimal α_2 for which the expected payoff for A_2 is higher than that for A_1 for all A_1, Z_1. Since the result holds $\forall\ Z_1$, we consider an environment where the cost of mining is negligible because in this environment, the adversary A_1 can make an arbitrary number of queries without incurring additional cost and is best suited for A_1 because $q \leq q^*$. Further, since the condition should be true $\forall\ A_1$, we take $\lim_{\alpha_1 \to \infty}$, using which we get the condition

$$(p_{fr} + \Delta) \frac{1 - \vartheta^{\alpha_2}}{1 - \vartheta} > p_{fr} \frac{1 - \vartheta^{\alpha_2}}{1 - \vartheta} + \frac{p_{fr} \vartheta^{\alpha_2}}{1 - \vartheta}$$

Let α_{opt} be the minimum α_2 that satisfies the above condition

$$\Delta \frac{1 - \vartheta^{\alpha_2}}{1 - \vartheta} \geq \frac{p_{fr} \vartheta^{\alpha_2}}{1 - \vartheta} \Rightarrow 1 - \vartheta^{\alpha_2} > \frac{p_{fr}}{\Delta} \vartheta^{\alpha_2}$$

$$\Rightarrow \vartheta^{\alpha_2} < \frac{\Delta}{p_{fr} + \Delta} \Rightarrow \alpha_2 log(\vartheta) < log(\frac{\Delta}{p_{fr} + \Delta})$$

Since $log(\vartheta) < 0$, and we want to minimize α_2, we can say

$$\alpha_{opt} = 1 + \lfloor \frac{log(\frac{\Delta}{p_{fr} + \Delta})}{log(\vartheta)} \rfloor$$

$$E[\mathcal{R}_{A_2}] = \Lambda\theta r_{block} p_{attack} \frac{1 - \vartheta^{\alpha_{opt}}}{1 - \vartheta} - \Lambda\alpha_{opt}\chi$$

$$E[\mathcal{R}_{A_1}] = \Lambda\theta r_{block} p_{fr} \sum_{i=0}^{\alpha_1} \vartheta^i - q^*\chi$$

$$E[\mathcal{R}_{A_1}] = \Lambda\theta r_{block} p_{fr} (\sum_{i=0}^{\alpha_{opt}} \vartheta^i + (\vartheta^{\alpha_{opt}} \sum_{i=0}^{\alpha_1 - \alpha_{opt}} \vartheta^i)) - q^*\chi$$

Step 3: Let us take simulator S_2 and environment Z_2 for which $\alpha_2 = \alpha_{opt}$. Let $D = \mathcal{R}_{A_2} - \mathcal{R}_{A_1}$. We also lower bound the difference $(q^* - \Lambda\alpha_{opt})\chi = 0$. So$E[D]$ is lower bounded by

$$E[D] \geq \Lambda\theta r_{block} \Big(\frac{1 - \vartheta^{opt}}{1 - \vartheta} (p_{attack} - p_{fr}) + \frac{\vartheta^{opt} p_{fr} (1 - \vartheta^d)}{1 - \vartheta} \Big)$$

Let $d = \alpha_1 - \alpha_{opt}$ ($d \geq 0$) and $\Delta = p_{attack} - p_{fr}$.

$$\Rightarrow E[D] \geq \Lambda\theta r_{block} \Big(\frac{1 - \vartheta^{\alpha_{opt}}}{1 - \vartheta} \Delta - \vartheta^{\alpha_{opt}} p_{fr} \frac{1 - \vartheta^d}{1 - \vartheta} \Big) \geq 0$$

The last inequality follows from the definition of α_{opt}. Since we solved this for arbitrary A_1, Z_1 (by showing result is true $\forall\ d \geq 0$), we conclude that A_2 is such that S_2, which simulates it in environment Z_2 has no $A_1 \in A_{fr}^{\infty}$ such that (S_1, Z_1) achieves higher payoff for any Z_1.

Further, we assumed that $\Delta > 0$, so even if there exists an attack that gives an adversary a slightly higher mining probability (e.g., SELFISH MINING WITH BRIBING), the protocol is not strongly attack-payoff secure. □

C.3 Proof for theorem 6.4

Proof. In this theorem, we show that any deflationary block-reward based cryptocurrency is attack-payoff secure against the set of adversarial strategies bound to α_{opt} rounds. That is, $\forall\ A \in \mathcal{A}^{\alpha_{opt}}$

$$\alpha_{opt} = 1 + \lfloor \frac{log(1-p_{fr})}{log(\vartheta)} \rfloor$$

Consider any adversary $A_2 \in \mathcal{A}^{\alpha_{opt}}$ with environment Z_2 where it makes $Q = q$ queries, for $q = maxsupport(P_Q)$. P_Q, Q are as explained in proof for Lemma 6.2. In this case, let q queries mine blocks such that in expectation α_2 phases are completed for $\alpha_2 < \alpha_{opt}$. The payoff for the adversary R_{A_2} is upper bounded by taking the probability of each query leading to a block being mined as 1.

$$R_{A_2} \leq \Lambda r_{block}\theta \sum_{i=0}^{\alpha_2 - 1} \vartheta^i = \Lambda\theta r_{block}\frac{1-\vartheta^{\alpha_2}}{1-\vartheta}$$

Consider a front-running semi-honest adversary A_1 and an environment Z_1 where the adversary makes q^* queries before the environment halts. In this case, consider p_{fr} be the probability of each query being accepted. Consider q^* such that it runs for α_1 rounds in expectation. The relation be such that $(1-\vartheta^{\alpha_1}) \geq \frac{(1-\vartheta^{\alpha_{opt}})\kappa}{(1-\delta)(1-\eta_{max})p_{min}}$, where all terms are same as defined in Step 1b in Appendix C.1. In this case, the adversary A_1 has payoff

$$R_{A_1} \geq \Lambda\theta r_{block} \sum_{i=0}^{\alpha_1 - 1} \vartheta^i - \Lambda\theta r_{block}\frac{1-\vartheta^{\alpha_1}}{1-\vartheta}$$

$$P[R_{A_1} < R_{A_2}] \leq P[\Lambda\theta r_{block} p_{fr} \tfrac{1-\vartheta^{\alpha_1}}{1-\vartheta} < \Lambda\theta r_{block}\tfrac{1-\vartheta^{\alpha_2}}{1-\vartheta}]$$

$$= P[p_{fr}\frac{1-\vartheta^{\alpha_1}}{1-\vartheta} < \frac{1-\vartheta^{\alpha_2}}{1-\vartheta}] = P[\frac{p_{fr}}{1-\vartheta^{\alpha_2}} < \frac{1}{1-\vartheta^{\alpha_1}}]$$

There always exist such α_1 for each $\alpha_2 < \alpha_{opt}$, this probability is 1 for some Z_1. (notice that we calculated α_{opt} in in C.2 to ensure this holds true). □

D Other Details

D.1 Reward Mechanisms in Blockchains

Blockchain Reward mechanisms have been studied in prior works Karakostas et al. [2022], Badertscher et al. [2018]. These reward mechanisms can be broadly categorized into two categories (1) Block Reward Mechanism (BRM) and (2) Transaction Fee Only Mechanism (TFOM).

Block Reward Mechanism (BRM). In the block-reward mechanism, the incentive for mining is provided through a special'coinbase' transaction. The cryptocurrency is deflationary if the block reward reduces every finite number of blocks mined, such that the sum of the net block reward is constant. In Bitcoin, Block-Reward halves every 210000 block mined. The payoff from transaction fees in BRM is very small compared to Block Reward and does not lead to strategic deviations.

Transaction Fee Only Mechanism (TFOM). In the TFOM, miners get negligible block rewards, and the main source of the payoff is the transaction fees from the transactions included in the blockchain.

D.2 Difference Utility

In this section, we explain the advantage of Protocol descriptors having their utility modeled a difference in utility of deviating and non-deviating parties.

Modeling utility in this way allows us to capture cases where the adversary does not gain a significant increase in payoff, but it reduces the payoff of other parties per round. Doing so allows the adversary to capture more fraction of the block reward from a particular phase. Further, the representation of utility as the difference between payoffs of deviating and non-deviating parties is also a more practical representation of the goal of protocol descriptor which is to minimize the benefit that a party gets from deviating.

D.3 Goldfinger Attack

Goldfinger attack Bonneau [2018] is one of the attacks where the adversary holds a short-position of the cryptocurrency and then launches a security attack. In this case, they profit from the decrease in the conversion rate of the cryptocurrency. This can be modeled in the payoff of the adversary as

$$v_A = \theta(t)E[R_B] + (\theta_{init} - \theta(t))c_1$$

Here θ_{init} is the coin's conversion rate when the short position was initially held. Therefore, as $\theta(t)$ decreases, the second term increases. This was not modeled in the utility structure of previous works such as Badertscher et al. [2018]. Modeling this allows us to argue that even if mining is not profitable, the adversary can have a positive payoff by shorting the crypto-currency.