TOWARDS RATIONAL CONSENSUS IN HONEST MAJORITY

A PREPRINT

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

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

May 13, 2024

ABSTRACT

Distributed consensus protocols reach agreement among n players in the presence of f adversaries; different protocols support different values of f. Existing works study this problem for different adversary types (captured by threat models). There are three primary threat models: (i) Crash fault tolerance (CFT), (ii) Byzantine fault tolerance (BFT), and (iii) Rational fault tolerance (RFT), each more general than the previous. Agreement in repeated rounds on both (1) the proposed value in each round and (2) the ordering among agreed-upon values across multiple rounds is called Atomic BroadCast (ABC). ABC is more generalized than consensus and is employed in blockchains.

This work studies ABC under the RFT threat model. We consider t byzantine and k rational adversaries among n players. We also study different types of rational players based on their utility towards (1) liveness attack, (2) censorship or (3) disagreement (forking attack). We study the problem of ABC under this general threat model in partially-synchronous networks. We show (1) ABC is impossible for $n/3 < (t + k) < n/2$ if rational players prefer liveness or censorship attacks and (2) the consensus protocol proposed by Ranchal-Pedrosa and Gramoli cannot be generalized to solve ABC due to insecure Nash equilibrium (resulting in disagreement). For ABC in partially synchronous network settings, we propose a novel protocol pRFT(practical Rational Fault Tolerance). We show pRFT achieves ABC if (a) rational players prefer only disagreement attacks and (b) $t < \frac{n}{4}$ and $(t+k) < \frac{n}{2}$. In pRFT, we incorporate accountability (capturing deviating players) within the protocol by leveraging honest players. We also show that the message complexity of pRFT is at par with the best consensus protocols that guarantee accountability.

Keywords distributed consensus, blockchains, security, equilibrium analysis

1 Introduction

Agreement and Distributed Consensus is a well-studied problem since its introduction Pease et al. (1980); Lamport et al. (1982a) as the Byzantine Generals' Problem. Applications include maintaining distributed file systems, building fault-tolerant systems, and, most recently, in Blockchain technology. Consensus is reaching agreement on a common value v among a set of players n with f faulty players. In case of repeated consensus, we require an additional condition that ordering among the agreed-upon values is the same across rounds. This generalization of agreement is called *Atomic BroadCast* (ABC).

Prior works achieved consensus in the presence of up to $t < n/3$ Byzantine failures Lamport et al. (1982a); Castro and Liskov (1999) under synchronous network assumptions. Castro and Liskov (1999) extended consensus to partially synchronous network through pBFT protocol. FLP Impossibility Fischer et al. (1985) stated that agreement through a *deterministic* protocol in asynchronous settings is impossible in the presence of even one faulty player. Later, randomized protocols were proposed Cachin et al. (2000, 2001); Bracha (1987), which achieve consensus in asynchronous network settings for $t < n/3$. Different consensus protocols work under different threat models. For instance, Paxos Lamport (1998, 2001) and Raft Ongaro and Ousterhout (2014) achieve consensus in presence of c *crash fault* players (where players can go offline). This threat model is Crash Fault Tolerant – $CFT(c)$. pBFT Castro and Liskov (1999), and

Network	Threat Model		
	$CFT(c)$	$BFT(t)$	$RFT(t,k)$
Synchronous	$2c < n$ Lamport (1998)	$2t < n$ Pease et al. (1980)	$t < \frac{n}{2}, k < \frac{n}{2}$ Pease et al. (1980)
Partially-synchronous	$2c < n$ Lamport (1998)	$3t < n$ Castro and Liskov (1999)	$t < \frac{n}{4}, t + k < \frac{n}{2}$
Asynchronous	$c < \frac{n}{3}$ Bracha (1987)	$t < \frac{n}{3}$ Bracha (1987)	$t < \frac{n}{3}$ Bracha (1987)

The results highlighted in blue are contributions of our work.

Table 1: Bounds for consensus in different threat models.

Honeybadger Miller et al. (2016) achieve consensus in the presence of $t < \frac{n}{3}$ byzantine faults (where the player can follow any arbitrary strategy). This is the Byzantine Fault Tolerant threat model – $BFT(t)$.

Ranchal-Pedrosa and Gramoli Ranchal-Pedrosa and Gramoli (2022) introduced a general rational threat model where t byzantine players and k rational players can collude. This is called a rational threat model – $RFT(k,t)$ and the agreement problem called Rational Consensus (RC). The authors propose RC using *baiting based protocol* – TRAP by showing the existence of a Nash Equilibrium (NE) that achieves consensus for $t < n/3$ and $k + t < n/2$. Protocols are called *Nash Incentive Compatible* (NIC) when the honest strategy is NE. However, we show the existence of another (more preferred) NE strategy that causes disagreement for TRAP when used to solve Atomic Broadcast (ABC). The rational players may prefer this dystopic equilibrium point over the more improbable secure equilibrium, making the protocol insecure. Game-theoretic security under the existence of multiple Nash equilibrium points is realized when following the protocol is Pareto-optimal/Focal equilibrium Schelling (1963) [1]. Protocols ensure stronger security guarantees if the equilibrium is Dominant strategy equilibrium (DSE) instead of Nash equilibrium (NE).

There is an absence of protocols realizing ABC in the rational threat model. Our work addresses this gap and proves impossibilities and a novel protocol pRFT that achieves ABC in the rational threat model under certain conditions on rational players' utility.

Our Approach

This work generalizes the $RFT(k,t)$ model Ranchal-Pedrosa and Gramoli (2022): (i) to incorporate payoff in repeated rounds with discounting and (ii) to model rational different agent types. These rational player types are depicted by θ: (1) $\theta = 1$ is incentivized towards disagreement, (2) $\theta = 2$ is incentivized towards censorship attack, and (3) $\theta = 3$ is incentivized towards denial of service (liveness is compromised). We show that for rational players types $\theta = 2, 3$, achieving Rational Consensus (RC) is not possible under the $RFT(t,k)$ threat model with $\frac{n}{3} \leq t + k < \frac{n}{2}$. Hence, we focus on rational players of type $\theta = 1$. We propose pRFT, which achieves consensus in $RFT(t,k)$ threat model for $\frac{n}{3} \leq t + k < \frac{n}{2}$ when rational players are of type[2] $\theta = 1$. Previous protocols attempted to incentive engineer the protocol such that rational players are incentivized to "bait" the deviating collusion. The rational players are incentivized to bait. The baiting is an equilibrium for the rational players if a certain threshold number (m) of players bait, leading to the protocol's security. However, this method was susceptible to the existence of alternate, insecure equilibrium points.

We capture deviation without relying on rational players (by incorporating accountability within the protocol), guaranteeing the capture of deviating players, making them prone to penalty. In our proposal, each player deposits some collateral. If deviation is captured, deviating players lose the collateral, which a rational player does not want. Hence, it will deviate if its collateral is intact. It leads to following the protocol as a dominant strategy for all rational players. Thus, our protocol achieves a stronger game-theoretic security guarantee, namely, Dominant Strategy Incentive Compatibility (DSIC). Based on these results, we place our work (coloured blue) in Table 1 among other known bounds for consensus in different network settings and threat models. We also show that pRFT achieves message complexity and message size, at par with the best available message complexity amongst protocols that guarantee *accountability* such as Ranchal-Pedrosa and Gramoli (2020); Civit et al. (2021) and pRFT works under a more general threat model than these protocols.

1.1 Our Contributions

In this paper, we first extend the Rational Threat Model proposed in Ranchal-Pedrosa and Gramoli (2022). We classify the rational players into three types represented by different values of θ (representing player types). $\theta = 3$ is when

[1] for details, refer to discussion in Section 4.3 or Schelling (1963)

[2] for details on different types of rational players, refer to Section 4.1.1

rational players are incentivized to compromise liveness and cause censorship or disagreement. $\theta = 2$ is when rational players are incentivized only to cause censorship or disagreement, and $\theta = 1$ is when players are incentivized only to cause disagreement. Based on this for k rational, t byzantine players such that $t < t_0$ and total players are n, we present the following impossibilities in Section 4.4.

- consensus is not possible when the set of rational players are of type $\theta = 3$ for $k + t < n/2$ and $t_0 < n/3$ in partially synchronous and asynchronous settings (Theorem 1).
- consensus is not possible when the set of rational players are of type $\theta = 2$ for $k + t < n/2$ and $t < n/3$ in partially synchronous and asynchronous settings (Theorem 2).
- There exists an additional Nash equilibrium that results in disagreement in baiting-based consensus protocols (such as TRAP, proposed in Ranchal-Pedrosa and Gramoli (2022)) (Theorem 3). Thus, there is a need for a new agreement protocol with one equilibrium point (preferably guaranteeing a stronger notion of Dominant Strategy Equilibrium instead of Nash Equilibrium[3]).

Following this, we propose a novel protocol pRFT (Section 5) which achieves consensus in $RFT(k, t)$ threat model. We show the following results for pRFT.

- pRFT achieves consensus with $k + t < n/2$ and $t < n/4$ when rational players are of type $\theta = 1$.
- pRFT guarantees correctness (Dominant Strategy Equilibrium) and liveness in synchronous and partially synchronous network settings.
- We show that pRFT achieves optimal message complexity among consensus protocols that provide accountability[4].

2 Related Work

The domain of distributed consensus has had extensive research in the past 50 years. We discuss below the work which is closely related to our work.

Byzantine Agreement. The inception of Byzantine Consensus saw the formulation of protocols under synchronous network settings Pease et al. (1980); Lamport et al. (1982a). This foundational work was subsequently extended to encompass partially synchronous scenarios Dwork et al. (1988). In the context of an asynchronous network, Fischer et al. Fischer et al. (1985) introduced the FLP impossibility by which it is impossible to reach an agreement using a *deterministic* protocol in the asynchronous network in the presence of even one faulty process. To overcome this, randomized protocols for distributed agreement and broadcast Cachin et al. (2000, 2001) in asynchronous settings were introduced. Blockchain technology introduced through Nakamoto's seminal whitepaper Nakamoto (2009) solves the State Machine Replication (SMR) using alternate protocols such as Proof-of-Work (PoW) and Proof-of-Stake (PoS) for consensus in public settings and BFT based Atomic Broadcast/Agreement (ABA) in the permissioned setting.

Atomic BroadCast (ABA) Protocols achieving ABA were introduced through Paxos Lamport (1998, 2001) and Raft Ongaro and Ousterhout (2014) which assumed a conservative *crash-fault* threat model and synchronous network assumptions. Deterministic protocols as pBFT Castro and Liskov (1999), Hotstuff Abraham et al. (2020); Yin et al. (2019), FlexibleBFT Malkhi et al. (2019) and others Civit et al. (2021); Aublin et al. (2013) achieved ABA in byzantine threat model. However, these protocols worked in synchronous and partially synchronous networks. Randomized ABA protocols such as Honeybadger Miller et al. (2016), Tardigrade Blum et al. (2021) and others Gao et al. (2022); Lu et al. (2022); Gilad et al. (2017); Spiegelman et al. (2022); Zhang and Duan (2022) achieve agreement even in asynchronous settings. The threat model used by these protocols is the byzantine threat model, which contains t byzantine adversaries. While in synchronous network agreement protocols Blum et al. (2021); Abraham et al. (2020) tolerate $n > 2f$, in partially-synchronous and asynchronous network agreement protocols Blum et al. (2021); Miller et al. (2016) tolerate $n > 3f$.

Rational Agreement The Byzantine threat model used to analyze distributed cryptographic protocols was extended by adding rational players. Aiyer et al. Aiyer et al. (2005) introduced the BAR (Byzantine Altruistic Rational) framework, where players/processes are altruistic, byzantine and rational. Rational players deviate only if utility from deviating is more than following the protocol honestly. BAR model has been since used to solve the distributed cryptographic problem called Transfer Problem Vilaça et al. (2011); Vilaça et al. (2011), which they solve when the producer of

[3] for details regarding notions of equilibrium points, see Section 4.3

[4] see Section 5.3.1 for a formal definition of accountable consensus protocols

N processors is such that $N > 2f$ for f (byzantine) faulty processes. Other works Abraham et al. (2006); Asharov et al. (2011); Badertscher et al. (2021); Garay et al. (2013); Ganesh et al. (2022) analyze the security of distributed cryptographic and game theoretic protocols against rational threat model. Some distributed protocols implement trusted third-party mediators using *cheap talks*. This process of realizing cheap talks secure against k rational and t byzantine adversaries for $k + 2t < n$ in synchronous settings Abraham et al. (2006) and $3(k + t) < n$ in asynchronous settings Abraham et al. (2019) similar to consensus protocols under byzantine adversaries Katz and Koo (2006); Blum et al. (2021). Groce et al. (2012) proposed analysis of agreement in the presence of two types of players — honest and rational adversaries. Ranchal-Pedrosa and Gramoli (2022) proposed TRAP protocol, which achieves rational consensus in the partially-synchronous and asynchronous network when $2(k + t) < n$ and $3t < n$. However, their result on the sufficiency of TRAP in achieving RC relies on rational players opting for an optimistic (but less plausible) equilibrium point instead of a dystopic (but more realistic) equilibrium, as demonstrated in this work. To our knowledge, the threat model discussed in Ranchal-Pedrosa and Gramoli (2022) is the most general present in literature, where t byzantine and k rational players exist, allowing arbitrary collusion among them. Our work extends this model by generalizing the type of rational player and presenting various exciting results in this setting.

3 Preliminaries

In this section, we motivate some definitions and prior works which are relevant to our work.

3.1 Blockchain and Agreement

Blockchain technology achieves *Atomic Broadcast* (ABC) (formally defined in Section 3.2), i.e. repeated consensus while preserving the total ordering of agreed-upon values. The probabilistic agreement is reached in permissionless settings using protocols such as PoW Nakamoto (2009) and PoS Gilad et al. (2017). Permissioned blockchains use BFT type of consensus where a committee (comprised of a set of players $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2, \ldots, \mathcal{P}_n\}$) proposes and achieves agreement on a single value in each round. In the context of Blockchain, the agreed-upon value is a Block. The block comprises *state-changes* to the global state of the system in the form of transactions. Each block B has a set of transactions $\overline{tx}$ and points to the parent block, i.e. the block agreed upon immediately before it.

The result is a chain of agreed-upon blocks, represented by C. Each player $\mathcal{P}_i$ has their own version of this chain, represented by C_i. Castro and Liskov (1999); Miller et al. (2016) wait for all nodes to agree on the same value. Other solutions like Gilad et al. (2017) and pRFT (our solution) partially confirm the blocks subject to rollbacks in case of adversarial behaviour. These blocks are labelled *Tentative Blocks*. Such blocks might be rolled back once the network synchronizes. They are considered finalized only if followed by a finalized block (defined in Gilad et al. (2017) as a block mined during a phase of synchrony). If the last *final block* was mined z blocks before the most recent block, then *common-prefix property* Garay et al. (2015a) holds if $\cap_{i=1}^{n} C_i^{\lfloor z}$ (i.e. chain obtained by removing the z most recent blocks in each player $\mathcal{P}_i$'s chain) is a prefix of all C_i.

Player Types We briefly discuss the type of players that are a part of our discussion. The system consists of n players, out of which h are honest, t are byzantine and k are rational.

- **Honest Players:** Also called altruistic players, they follow the specified protocol honestly.
- **Byzantine Players:** They follow any strategy with the intent to cause disruption in the correctness, liveness or other properties guaranteed by the distributed protocol. They are immune to incentive manipulation and will choose a strategy irrespective of their *payoff* from that strategy.
- **Rational Players:** These players follow the strategy which gives the highest *payoff* based off of some utility structure (which is protocol and agent type specific[5]). Therefore, such players deviate from following the protocol honestly only if there exists a strategy with a higher payoff.

3.2 Flavours of Consensus

The problem of consensus in a distributed setting was first motivated by the Byzantine General's problem Lamport et al. (1982a) and has since been discussed in the literature in different capacities such as Byzantine Broadcast (BB), Byzantine Agreement Pease et al. (1980); Lamport et al. (1982a) (BA), Atomic Broadcast Cristian et al. (1995); Blum et al. (2021) (ABC) and Rational Consensus Aiyer et al. (2005); Ranchal-Pedrosa and Gramoli (2022) (RC). In this

[5]for more details on utility structure in our case, refer to Section 4.1.2

section, we will discuss notions of consensus that serve as preliminaries to our work. We elaborate some of the more common definitions in Appendix A.

3.2.1 Byzantine Broadcast & Agreement

Byzantine Generals' Problem was introduced by Lamport et al. Pease et al. (1980). In the Byzantine Agreement problem, the system is comprised of t faulty (byzantine) and $n - t$ non-faulty (honest/altruistic) players. We motivate the definition of BA from (Blum et al., 2021, Definition 2).

3.2.2 Atomic Broadcast

Atomic Broadcast (ABC) is a generalization of BA. In BA, players reach an agreement on a value v once, while in ABC, players reach this agreement multiple times, maintaining a ledger of agreed-upon values. ABC is therefore repeated rounds of agreement on values such that a ledger is maintained with an added constraint that the ordering of different values is the same for all honest players. The formal definition of ABC is motivated from (Blum et al., 2021, Definition 5).

3.2.3 Rational Consensus

With the increasing interest in the rational security analysis of distributed cryptographic protocols, we motivate from Ranchal-Pedrosa and Gramoli (2022) and define *Rational Consensus* (RC) – the equivalent of ABC with a general (byzantine and rational) threat model as follows. We motivate the definition of robustness from Ranchal-Pedrosa and Gramoli (2022) and extend it to repeated rounds by adding the condition of c-strict ordering (the rational equivalent of ABC).

Definition 1 ((t, k)-robustness)**.** Consider a protocol Π is run by $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2 \ldots, \mathcal{P}_n\}$ players where t players are byzantine and k players are rational (follow the strategy with the highest incentive) while remaining $n - t - k$ players are honest. The protocol Π is (t, k)-robust if it satisfies:

- (t, k)-validity If all altruistic players receive value v then they all agree on value v
- (t, k)-agreement All altruistic players output the same value in each round.
- $c-$strict ordering If the ledger of agreed blocks is C_1 and C_2 for two altruistic players with $|C_1| \leq |C_2|$, then $C_1^{\lfloor c} \subseteq C_2^{\lfloor c}$ holds[6].
- (t, k)-eventual liveness if an honest player outputs B then all honest players output B eventually.

We define a stronger notion of RC when protocols also satisfy censorship resistance (we define (t, k)-censorship resistance) and call such RC protocols as strongly (t, k)-robust.

Definition 2 ((t, k)-censorship resistance)**.** A protocol Π satisfies (t, k)-censorship resistance if when all honest players have transaction tx as input, then eventually all honest players output a block with transaction tx.

Definition 3 (strong (t, k)-robustness)**.** A protocol Π is strongly (t, k)-robust if Π is (t, k)-robust and (t, k)-censorship resistant.

3.3 Cryptographic and Network Preliminaries

Digital Signatures We employ the use of digital signatures and assume unforgeability except with negligible probability by all players (players are polynomially bounded) with access to random oracle[7]. This use of PKI (Public Key Infrastructure) for unforgeable digital signature has been employed for Authenticated Byzantine Agreement, first introduced by Dolev & Strong Dolev and Strong (1983).

Trusted Setup Before initiation of the protocol, we assume there is a trusted broadcast-type setup similar to Cachin et al. (2000) (implemented via a common third party) where all participating players share their public keys, against which any digitally signed message is verified.

[6] $C^{\lfloor c}$ is the ledger after removing the last c blocks

[7] such players represent the set of all Probabilistic Polynomial Time Turing Machines (PPTM)

Network Settings We assume reliable channels between each pair of players involved in our analysis. Therefore, messages cannot be lost or tampered with, but they can face network delays. Based on the dealy, we consider three types of networks: (1) *synchronized* is when the delay is upper bounded by a known bound Δ which can be used to parameterize the protocol. (2) *asynchronous* network does not have an upper bound on the delay, but the message eventually gets delivered (i.e. delay for each message is finite). (3) *partially-synchronous* Dwork et al. (1988) network is when the system behaves as an asynchronous network till before an event called *Global Stabilization Time* (GST), after which the system becomes synchronous with some upper bound on delay.

3.4 Baiting based Consensus Protocols

Baiting-based consensus protocols such as Ranchal-Pedrosa and Gramoli (2022) assume collusion of $k + t$ players (k rational and t byzantine) deviating from the agreement protocol. They employ a baiting strategy to incentivize rational players to bait the collusion by submitting Proof-of-Fraud, which consists of $t_0 + 1$ conflicting signatures (for a detailed discussion on proof-of-fraud see Section 5.3.1 and Civit et al. (2021); Ranchal-Pedrosa and Gramoli (2020)). If m rational players follow the baiting strategy, then one of them is randomly selected for the reward $\mathcal{R}$ associated with baiting. Each player has a deposit $\mathcal{L}$, which they lose if there is Proof-of-Fraud containing their conflicting signatures. Additionally, there is a net utility gain of $\mathcal{G}$ for the collusion $K \cup T$ if the system ends up in disagreement. This utility is equally divided between the set of rational players such that each player $P_i \in K$ gets $\frac{\mathcal{G}}{k}$ payoff. For a more extensive description of the system and results used in TRAP, refer to Ranchal-Pedrosa and Gramoli (2022).

4 Our Model

We model RC as a game between three types of players Byzantine, Rational and Altruistic (Honest) Players. In this section, we define (i) the game, (ii) the utility structure and (iii) network models.

4.1 The Game

The Game consists of a set of players $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2 \ldots, \mathcal{P}_n\}$ who maintain a ledger of Blocks. Each Block contains a set of transactions $\overline{tx} = (tx_i)_{i=1}^{z}$ which are valid wrt. previously *confirmed* blocks. Agreement on a single Block proceeds in discrete intervals called *rounds*. In each round r we have a leader $\mathcal{P}_l$ (for $l = 1 + (r \bmod n)$) that proposes a block B_r.

4.1.1 Players

We have three types of players: Byzantine, Rational and Honest players.

- **Byzantine:** There are t byzantine players belonging to set $T \subset \mathcal{P}$. They follow any arbitrary strategy irrespective of payoff, with the goal of causing maximum disruption of the system.
- **Rational:** There are k rational players belonging to the set $K \subset \mathcal{P}$ which follow the strategy that provides them maximum utility. They follow the protocol Π honestly unless there exists a deviation that gives them more than *negligible* advantage in utility. The rational players can be of one of four types, represented by $\theta \in \{0, 1, 2, 3\}$.
- **Honest:** There are $h = n - k - t$ Honest players belonging to the set $H \subseteq \mathcal{P}$. These players follow the protocol honestly as long as participation in the protocol is incentivized over abstaining from participation, otherwise, these players don't participate in the protocol.

To allow a larger attack space for the adversary, we consider that the Rational and Byzantine players can collude with each other. Therefore, there can exist a collusion set $\subseteq K \cup T$ of size $\leq k + t$.

System States Due to strategies followed by the players in the system and due to the external environment (network delays), a distributed system can be in the following states:

- **No Progress (σ_{NP}):** In any round $r(\forall r \in \mathbb{R})$ no new blocks are mined.
- **Conditional Progress (σ_{CP}):** In any round $r(\forall r \in \mathbb{R})$ the confirmed blocks contain transactions such that $\forall tx_i \in \overline{tx}, tx_i \notin Z$ where Z is the set of *censored transactions*.
- **Disagreement (σ_{Fork}):** In any round $r(\forall r \in \mathbb{R})$ we have two honest players $\mathcal{P}_i, \mathcal{P}_j \in H$ such that their ledger state has two *confirmed blocks* B_i and B_j at the same height h and $B_i \neq B_j$.
- **Honest Execution (σ_0):** In any round, the protocol executes according to the honest execution and does not violate correctness or liveness conditions.

Player Type (θ)	System State (σ)				Preferred States
	σ_{NP}	σ_{CP}	σ_{Fork}	σ_0	
$\theta = 3$	α	α	α	0	No Progress, Censorship, Fork
$\theta = 2$	$-\alpha$	α	α	0	Censorship, Fork
$\theta = 1$	$-\alpha$	$-\alpha$	α	0	Fork
$\theta = 0$	$-\alpha$	$-\alpha$	$-\alpha$	0	Honest Execution

Table 2: Payoff function $f(\sigma, \theta)$

Player Types We further model rational player type[8] through $\theta \in \{0, 1, 2, 3\}$. The players' types depend on their incentives for different states of the distributed system. We characterize the function $f(\sigma, \theta)$ for payoff when player type is θ and the system is in state σ. This function is represented in Table 2 (for some positive constant α). Types of rational players are described below:

- $\theta = 3$: Such players are incentivized if the system state is σ_{NP}, σ_{CP} or σ_{Fork}.
- $\theta = 2$: Such players are incentivized if system state is σ_{CP} or σ_{Fork}.
- $\theta = 1$: Such players are incentivized if system state is σ_{Fork}.
- $\theta = 0$ This type of rational player is disincentivized if the system is in any state except σ_0. However, they might be incentivized against sending messages or performing verification of messages, which has been previously analysed by Amoussou-Guenou et al. (2020).

If there are multiple types of rational players, we analyze for security for the worst types amongst them. If K_i is set of rational players with type $\theta = i$, we say, $K = \cup_{i=0}^{3} K_i$ is of type $\theta = \max\{i | K_i \neq \emptyset\}$.

4.1.2 Utility Structure

Byzantine players follow the strategy that causes maximal disruption to the protocol Π irrespective of the associated utility. If the protocol is *individually rational*, honest players follow Π honestly. Therefore, we need only model the utility for the rational players $\mathcal{P}_i \in K$.

Strategy Space In addition to defining the possible states of the system and the possible types of a rational player, we define the set of strategies available with the rational players in each round.

- **Abstain** π_{abs}**:** $\mathcal{P}_i$ does not send messages in the particular phase or round.
- **Double-Sign** π_{ds}**:** $\mathcal{P}_i$ signs on two conflicting messages in the same phase of the same round.
- **Honest** π_0**:** $\mathcal{P}_i$ follows the specified protocol Π honestly.

Penalty There also exists a penalty that is incurred by player $\mathcal{P}_i$ if there exists proof that with overwhelming probability, the player has deviated from the protocol. The penalty is a fixed constant L for each player which is the collateral deposited by these players before participating in consensus. If proof of malicious behaviour is found, this deposit is stashed/burnt Karantias et al. (2020) and the penalty mechanism should be such that for a player that has followed the protocol honestly the penalty should be 0 except with negligible probability.[9] The penalty is determined through a function $D(\pi, \sigma)$ based on the mechanism which takes value 1 if a penalty is incurred and 0 otherwise.

Based on the strategy π and type θ, we define the utility of rational players in a round r by taking an expectation over the set of possible states, i.e. $\sigma \in S$ as: $u_i(\pi, \theta, r) = \mathbb{E}_{\sigma \sim S}[f(\sigma, \theta)] - L \cdot D(\pi, \sigma)$

The function $f : S \times \{0, 1, 2, 3\} \to \{-\alpha, 0, \alpha\}$ (for some positive constant α) is given in Table 2. If we consider the expected utility in a particular round for player $\mathcal{P}_i \in K$ as $u_i(\pi, \theta, r)$, then the expected utility across rounds can be defined as:

$$U_i(\pi, \theta) = \sum_{r=0}^{\infty} \delta^r u_i(\pi, \theta, r) \tag{1}$$

[8]The notion of player type θ corresponds only to rational players because Byzantine type ($\theta = 3$) and Altruistic/Honest type ($\theta = 0$) is already fixed by definition.

[9]This condition ensures *Individual Rationality* of Honest players

4.2 Threat Model

We model threat via $\mathcal{M}$ where $|T| = t \leq t_0$ and $|K| = k$. Here t_0 is the upper-bound on byzantine players to ensure security against byzantine-only attacks.

We make a simple observation under the threat model $\mathcal{M}$, for any $t_0 \geq 1$, the necessary condition for any protocol Π to reach agreement with the threshold $\tau \in [\lfloor \frac{n+t_0}{2} \rfloor + 1, n - t_0]$. This threshold is such that $n - t_0$ players should agree on a value for agreement.

Claim 1. *A protocol Π achieves consensus with agreement of at least τ players agreeing on the same value under threat model $\mathcal{M} := \langle (\mathcal{P}, T, K), \theta, t_0 \rangle$ only if $\tau \in [\lfloor \frac{n+t_0}{2} \rfloor + 1, n - t_0]$*

Proof. We prove the contrapositive of this claim. Consider the two cases where $\tau > n - t_0$ and $\tau \leq \lfloor \frac{n+t_0}{2} \rfloor$. If $\tau > n - t_0$ then a message/vote from at least one byzantine player is required to reach a consensus. Under this situation, each byzantine player can play π_{abs} which would compromise the (t, k)-eventual liveness property of the protocol Π. If $\tau \leq \lfloor \frac{n+t_0}{2} \rfloor$ then consider the existence of network partition such that two subsets of players A and B are unable to communicate with each other except through set of adversaries T. Here, $A \cup B = \mathcal{P} \setminus T$, $A \cap B = \emptyset$ and $|A| = |B| = \frac{n-t_0}{2}$. If the leader in some round is $\mathcal{P}_l \in T$ then leader proposes v_a to A and v_b to B. Since $|A| + |T| \geq \lfloor \frac{n-t_0}{2} \rfloor + t_0 \geq \tau$ and similarly $|B| + |T| \geq \lfloor \frac{n-t_0}{2} \rfloor + t_0 \geq \tau$, both partitions reach consensus on conflicting values which invalidates the (t, k)-agreement property. Therefore, Π is not (t, k)-robust in either case. □

4.3 Equilibrium & Incentive Compatibility

Nash Equilibrium (NE) While analysing the rational security of protocols, a protocol is considered secure if following the protocol honestly is a *Nash Equilibrium* strategy. Therefore, the aim is to design a protocol that is Nash Incentive Compatible. This means that following the protocol honestly is the *nash-equilibrium* strategy for all rational players. We define Nash Incentive Compatible protocol as follows:

Definition 4 (Nash Incentive Compatible)**.** A protocol Π with set of K rational players is *Nash Incentive Compatible* (NIC) for a given utility structure U if $\forall i \in K$ in the set of rational players K following strategy s_i following the honest strategy π_0 is Nash Equilibrium. i.e. $\forall i \in K, \forall \pi$

$$U_i(s_i = \pi, \{s_j = \pi_0\}_{j \neq i}) \leq U_i(s_i = \pi_0, \{s_j = \pi_0\}_{j \neq i})$$

Focal Point Given a game, it could have multiple equillibria. Amongst, multiple equillibria, a particular equilibrium may attract more attention than other equllibria. Such equilibrium is often referred as *focal point* Schelling (1963). Consider the following example game between three players $P1$ having strategy space $\{A, B\}$, $P2$ having $\{a, b\}$ and $P3$ having $\{\alpha, \beta\}$. The utility is as given in Table 3 given in the order (U_{P1}, U_{P2}, U_{P3}).

	a		b	
	α	β	α	β
A	$(1, 1, 1)$	$(1, 1, 0)$	$(1, 0, 1)$	$(-2, 2, 2)$
B	$(0, 1, 1)$	$(1, -2, 1)$	$(2, 2, -2)$	$(0, 0, 0)$

Table 3: Example of two equilibria in a 3-player game (Here, utility is in the order (A, a, α))

The game has two Nash equillibria, — (B, b, β) and (A, a, α). The latter is attractive as it offers higher utility to all the players. Such focal points are important in analyzing a security game.

Challenges with multiple Nash Equilibria in a Security Game In case there are multiple NEs of a security game, if one of them implies security, does not imply security in general. The attackers would take the game towards an insecure equilibrium or rational players may play strategies resulting in equilibrium with higher utilities to them. Thus, we must explore all equlibiria and ensure security at the worst equilibrium.

In our analysis, we are going to argue that in the generalized model in this paper, there are multiple equilibria and baiting-based equilibrium Ranchal-Pedrosa and Gramoli (2022) that ensure the security of the protocol is one of them (similar to (B, β, b) in Table 3). We show in Theorem 3 that there is another equilibrium (similar to (A, α, a)) which may be more attractive to rational players. At the later equilibrium, the protocol is not secure. Thus, in the generalized model, TRAP Ranchal-Pedrosa and Gramoli (2022) need not be secure.

Dominant Strategy Equilibrium (DSE). A better equilibrium from the weaker *stable Nash equilibrium* is a DSE. DSE equilibrium points are not contested by other equilibrium points. Thus, we can safely assume rational agents follow a DSE strategy, and the protocol is Dominant Strategy Incentive Compatible (defined below).

Definition 5 (Dominant Strategy Incentive Compatible)**.** A protocol Π with a set of K rational players is *Dominant Strategy Incentive Compatible* (DSIC) for a given utility structure U if $\forall i \in K$ following honest strategy $s_i = \pi_0$ is Nash Equilibrium. i.e. $\forall i \in K, \forall \pi, \forall s_j \forall j \in K/\{i\}$

$$U_i(s_i = \pi, \{s_j\}_{j \neq i}) \leq U_i(s_i = \pi_0, \{s_j\}_{j \neq i})$$

We propose pRFT in Section 5 which is DSIC; therefore providing a better security guarantee.

4.4 Impossibilities

From the discussion of the previous section, rational players can be of different types $\theta \in \{0, 1, 2, 3\}$. We also know through Claim 1 that any protocol requires $\tau \in [\lfloor \frac{n+t_0}{2} \rfloor + 1, n - t_0]$ for security against *byzantine* attacks (Note that, this is necessary but not sufficient). We show through Theorem 1 that under a stricter (more adversarial) agent type for rational players, achieving RC for any $k + t > \frac{n}{3}$ is impossible.

Theorem 1 (Rational Consensus under $\theta = 3$)**.** *Under the threat model $\langle (\mathcal{P}, T, K), \theta = 3, t_0 \rangle$ no rational consensus protocol is (t, k)-robust when $\lceil \frac{n}{3} \rceil \leq k + t \leq \lceil \frac{n}{2} \rceil - 1$.*

Proof. The proof shows that for any arbitrary protocol Π, rational players are incentivized to follow π_{abs}. Since π_{abs} is indistinguishable from crash faults, no penalty-based (accountable) protocol can distinguish this behaviour (and thus reduce the utility). Due to space constraints, we omit the inclusion of the full proof in the paper. This compromises (t, k)-eventual liveness property and therefore, any protocol Π is not a (t, k)-robust RC protocol. □

We also show a pessimistic result for rational players of $\theta = 2$. We show through Theorem 2 that for every protocol if the rational players are of type $\theta = 2$ then there always exists a strategy which compromises (t, k)-censorship resistance while ensuring (t, k)-eventual liveness. Note that this impossibility holds despite of existence of threshold encryption schemes Miller et al. (2016),

Theorem 2 (Rational Consensus under $\theta = 2$)**.** *Under threat model $\langle (\mathcal{P}, T, K), \theta = 2, t_0 \rangle$ no rational consensus protocol is strongly (t, k)-robust when $\lceil \frac{n}{3} \rceil \leq t + k \leq \lceil \frac{n}{2} \rceil - 1$.*

Proof. The proof follows by showing the existence of a strategy π_{pc} where adversarial collusion set follows (1) π_{abs} when the leader is honest, (2) π_0 (and omit censored transactions) when the leader is adversarial. Due to the indistinguishability between crash faults and π_{abs}, no accountable protocol is possible. Due to space constraints, the full proof is provided in Appendix C. □

Having proven the impossibility of having a protocol that achieves RC for rational player types $\theta = 3$ and 2, we now relax the player type further to $\theta = 1$. Under this utility model for rational players, Gramoli et al. Ranchal-Pedrosa and Gramoli (2022) proposed a Baiting-based protocol (which they call TRAP) and show Baiting is *necessary* and *sufficient* to achieve RC under partially-synchronous network settings for $t + k < \frac{n}{2}$. Following our brief discussion of the model and result of Ranchal-Pedrosa and Gramoli (2022) in Section 3.4, we show that the Baiting-based protocol does not ensure RC in repeated rounds of consensus due to the existence of another Nash equilibrium point which leads to disagreement. Further, this point being focal equilibria (see discussion in Section 4.3), it will be preferred over the secure equilibria point proposed by Ranchal-Pedrosa and Gramoli (2022). We demonstrate this result in Theorem 3 below. For notational consistency of this result with Ranchal-Pedrosa and Gramoli (2022) we use $R := f(\sigma_{fork}, 1)$

Theorem 3 (Baiting based Rational Consensus under $\theta = 1$)**.** *Consider any baiting-based RC protocol Π the threat model $\mathcal{M} = \langle (\mathcal{P}, K, T), \theta = 1, t_0 \rangle$. The set of rational players following π_{fork} is a Nash-equilibrium for $|K| > 2 + t_0 - t$. Thus, Π is not (t, k)-robust RC for $t_0 = \lceil \frac{n}{3} \rceil - 1$.*

Proof. The proof follows from showing that if collusion follows grim-trigger strategy[10], then another Nash Equilibria (NE) exists in repeated rounds where players of the collusion deviate in every round. This Equilibrium is pareto-optimal due to which players are more prone to end up at this NE point than the secure NE proposed by Ranchal-Pedrosa and Gramoli (2022). □

[10] grim-trigger: if one player of collusion baits, all players will abandon collusion.

In this section, we demonstrated the impossibility of achieving RC when rational players are of the type $\theta = 3$ or 2 and $k + t > \frac{n}{3}$. We further show that when rational players are of type $\theta = 1$, the previously existing "baiting-based" solution Ranchal-Pedrosa and Gramoli (2022) has a Nash equilibrium strategy, resulting in Disagreement. From our discussion in Section 4.3, this is a more stable equilibrium point than the secure equilibrium point proposed by Ranchal-Pedrosa and Gramoli (2022) (that relies on $m > \frac{t+k-n}{2} + t_0$ deviating together; ref. Lemma 4.3 and Lemma 4.4 in Ranchal-Pedrosa and Gramoli (2022)). In the next section, we propose a protocol pRFT which achieves (t, k)-robust RC without relying on baiting by rational players.

5 pRFT: Rational Consensus Protocol

Baiting-based consensus Ranchal-Pedrosa and Gramoli (2022) introduces interesting insights about using Proof-of-Fraud (PoF) to penalize deviating rational players. The protocol relies on incentivizing rational players to bait the collusion, and we show in Theorem 3 existence of another (better) Nash Equilibrium for any rational player to deviate to π_{Fork} than follow honest-baiting π_{bait} strategy. To solve this problem, we attempt to capture PoF through honest players prone to any incentive manipulation. Towards this, we propose pRFT, which is described below.

5.1 Protocol

The pRFT protocol runs in discrete rounds. The set of players involved are $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2 \ldots, \mathcal{P}_n\}$ and in round r the leader is $\mathcal{P}_l$ for $l = 1 + (r \bmod n)$. We assume the network is partial-synchronous, and reliable-broadcast i.e. messages reach to the receiver untampered, although they might suffer network delays. Note that for brevity, we abstract the cryptographic verification of the message to be done by the Recv procedure (lines $7, 11, 17, 24, 27$ and 29 of the protocol in Figure 1). Therefore, any message coming through it will contain only valid signatures, and invalid messages are discarded. Each round progresses in 4 distinct phases, described as follows:

Propose Phase The leader $\mathcal{P}_l$ has a set of transactions that they want to publish to the ledger (blockchain). $\mathcal{P}_l$ selects a set of these transactions $\overline{tx} = \{tx_1, tx_2 \ldots, tx_c\}$ and form a Block B_r. She then proposes this block by broadcasting over the network via a *propose* message with their cryptographic signature s_l on the hash $h_l := \mathbf{H}(\mathsf{Block}||r)$ of the Block[11]. They construct message $m := (\langle Propose, B_l, h_l, r\rangle$ and signature s_l^{pro} to broadcast (m, s_l^{pro}). All non-leader players $\mathcal{P}_i$ receive the propose message from the leader and move to the vote phase. The non-leader players $\mathcal{P}_i$ $\forall i \neq l$ (1) check the validity of the *propose* message from the leader $\mathcal{P}_l$ (2) broadcast a *vote* message if the *propose* message is valid. In checking the validity of the *propose* message, the player verifies $H_l = \mathbf{H}(\mathsf{Block}||r)$ and verifies signature s_l on the message $\langle Propose, B_l, h_l, r\rangle$. They then sign s_i^{vote} on the message $m_i^{vote} := \langle Vote, h_i, , s_l^{pro}, r\rangle$ and broadcast (m_i^{vote}, s_i^{vote}) over the network.

Vote Phase In this phase, they wait for $n - t_0$ *valid* vote messages from other players from the previous phase for some (same) proposed value h_*. If no such value is obtained, the player sets $h_* := \perp$ (default empty value). Each player commits to the value h_* by constructing a Commit message. This consists of the decided value h_* and set V_i of $\geq n - t_0$ votes on this value $V_i = \emptyset$ if $h_* = \perp$. The player $\mathcal{P}_i$ signs on the message $m_i^{com} := \langle Commit, h_*, s_l^{pro}, V_i, r\rangle$ and broadcasts (m_i^{com}, s_i^{com}) to all other players.

Commit Phase Upon receiving $\geq n - t_0$ commit messages for a particular (same) value h_{tc}, the player reaches *tentative-consensus* on this block. Each player $\mathcal{P}_i$ shares their tentative consensus by sharing value h_{tc} which got $\geq n - t_0$ commit messages and a Proof-of-Commitment which is the set W_i (for player $\mathcal{P}_i$ of $n - t_0$ signatures on the commit messages on value h_*. They construct message $m_i^{rev} := \langle Reveal, h_{tc}, h_l, W_i, r\rangle$ and signature s_i^{rev} on this message to broadcast (m_i^{rev}, s_i^{rev}).

Reveal Phase Each player verifies across the set of Proof-of-Commitments W_j for each (m_j^{rev}, s_j^{rev}) received by $\mathcal{P}_i$ any attempts of *double-signature* in the proof vectors. Each player accumulates a set of double signatures as Proof-of-Fraud (PoF) by invoking the ConstructProof procedure (see Figure 4). These PoF can be used to *burn* the tokens/coins deposited by the deviating player by including a corresponding transaction in a future block. If there are $\geq n - t_0$ messages and a total of $\leq t_0$ players with double signatures (PoF) then the player reaches *Final Consensus* on the proposed block B_l. In this case, they construct a message $m_i^{fin} := \langle Final, h_l, s_l^{pro}\rangle$ and signature s_i^{fin} on it. They then broadcast this pair (m_i^{fin}, s_i^{fin}) to the client/network. Otherwise, if the set of double signatures $\geq t_0 + 1$, they broadcast this PoF with $\geq t_0 + 1$ double-signatures (represented by set D_i). They then construct message $m_i^{exp} := \langle Expose, D_i, r\rangle$ and signature s_i^{exp} and broadcast (m_i^{exp}, s_i^{exp}). If the player obtains $> \frac{n}{2}$ Final messages,

[11] h_l also contains the round number therefore, signed messages from one round can not be used in another round using replay attack.

pRFT$(\overline{p}_{i=1}^{n}, t_0)$

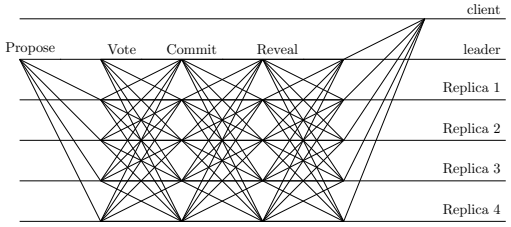
Propose Phase:
if $i = r \bmod n$ **then**
  Block$_r$:= ConstructBlock$(\overline{tx}, r, b_{parent}, p_i)$
  $h_l :=$ Hash(Block$_r$)
  Broadcast$(\langle Propose, Block_r, h_l, r\rangle_i, s_i^{pro})$
else
  On Recv. $(\langle Propose, Block_r, h_l, r\rangle, s_l^{pro})$:
    Broadcast$(\langle Vote, h_l, s_l^{pro}, r\rangle_i, s_i^{vote})$
end if

Vote Phase:
On Recv. $(\langle Vote, h_j, s_l^{pro}, r\rangle, s_j^{vote})$:
  $votes[h_j] := votes[h_j] \cup \{s_j^{vote}\}$
if for some $h_*, vote[h_*].size \geq n - t_0$ **then**
  Broadcast $(\langle Commit, h_*, s_l^{pro}, vote[h_*], r\rangle_i, s_i^{com})$
end if

Commit Phase:
On Recv. $(\langle Commit, h_j, s_l^{pro}, \overline{vote}_j, r\rangle, s_j^{com})$:
  $commit[h_j] := commit[h_j] \cup \{s_j^{com}\}$
if for some $h_*, commit[h_*].size \geq n - t_0$ **then**
  Broadcast$(\langle Reveal, h_*, s_l^{pro}, commit[h_*], r\rangle_i, s_i^{rev})$
end if

Reveal Phase:
$D_i := \emptyset, M_i := \emptyset, F_i := \emptyset$
On Recv. $(\langle Reveal, h_j, s_l^{pro}, \overline{commit}_j, r\rangle, s_j^{rev})$:
  $M_i \leftarrow M_i \cup \{\overline{commit}_j\}$
  $D_i :=$ ContructPoF(M_i)
On Recv $(\langle Final, B_j, s_l^{pro}\rangle_j, s_j)$
  $F_i \leftarrow F_i \cup \{s_j^{fin}\}$
On Recv $(\langle Expose, D_j, r\rangle, s_j)$
  Stash$(D_j), r := r + 1$
if $|D_i| > t_0$ **then**
  Broadcast$(\langle Expose, D_i, r\rangle_i, s_i^{expose})$
else if $|M_i| \geq n - t_0$ **then**
  Broadcast$(\langle Final, B_l, s_l^{pro}\rangle_i, s_i^{fin})$
else if $|F_i| > \frac{n}{2}$ **then**
  Broadcast$(\langle Final, B_l, s_l^{pro}\rangle_i, s_i^{fin})$
end if

Figure 1: pRFT Protocol



(a) Normal execution of pRFT protocol

Message Type	Message
propose	$(\langle Propose, B_l, h_l, r\rangle, s_l^{pro})$
vote	$(\langle Vote, h_i, , s_l^{pro}, r\rangle, s_i^{vote})$
commit	$(\langle Commit, h_*, s_l^{pro}, V_i, r\rangle, s_i^{com})$
reveal	$(\langle Reveal, h_{tc}, h_l, W_i, r\rangle, s_i^{rev})$
expose	$(\langle Expose, D_i, r\rangle, s_i^{exp})$
final	$(\langle Final, h_l, s_l^{pro}\rangle, s_i^{final})$
view-change	$(\langle ViewChange, \mathsf{Phase}, r, \rangle, s_i^{vc})$
commit-view	$(\langle CommitView, V_i, r\rangle, s_i^{cv})$

(b) Messages sent in pRFT

Figure 2: pRFT protocol execution & messages

this means at least one honest player has finalized on the block. Then, this player also finalizes and broadcasts a final message.

5.2 View Change

In each round players wait for proposal messages in the proposed phase or $\geq n - t_0$ messages in the other phases. Either due to delays in the network or $> t_0$ players deviating from the protocol, if a timeout happens (when the duration of phase exceeds the local waiting time of Δ) then *view change* is initiated. The view change protocol proceeds as follows:

1 A player triggers view-change if the following happens:

- a timeout in waiting time Δ
- conflicting signatures on 2 different proposed values v_{l1}, v_{l2} in the same round by the leader $\mathcal{P}_l$
- conflicting signatures by $\geq t_0 + 1$ players in some phase.

Under either of these scenarios, the player signs the message $m_i^{vc} := \langle View - Change, \mathsf{Phase}, r, \rangle$ with signature s_i^{vc} and broadcast (m_i^{vc}, s_i^{vc}) to the network.

Protocol	Message Complexity	Message Size	Accountability
pBFT Castro and Liskov (1999)	$O(n^3)$	$O(\kappa \cdot n^4)$	×
Hotstuff Yin et al. (2019)	$O(n^2)$	$O(\kappa \cdot n^3)$	×
Polygraph Civit et al. (2021)†	$O(n^3)$	$O(\kappa \cdot n^4)$	✓
pRFT	$O(n^3)$	$O(\kappa \cdot n^4)$	✓

†While polygraph achieves same guarantees, their threat model is much weaker than pRFT's

Figure 3: Message Complexity for different consensus protocols compared with pRFT

2 If a player $\mathcal{P}_i$ receives a view-change message from player $\mathcal{P}_j$ for some phase in some round, they store the message (1) Wait for $\geq n - t_0$ such messages (from the same phase) or timeout or, (2) If they have $\geq n - t_0$ messages from that phase, they send the corresponding messages to $\mathcal{P}_j$.

3 If a player receives $\geq n - t_0$ *view-change* messages including their own *view-change* (represent this set as V_i), they construct a *commit-view* message $m_i^{cv} := \langle CommitView, V_i, r \rangle$ along with the signature s_i^{cv}. The player will discontinue the current round and wait for a view change.

4 If a player receives a commit-view message, they verify if it consists of $n - t_0$ valid (signed) view-change messages. If so, they commit to view change and send a *commit-view* message.

5 When the player receives $> n - t_0$ commit-view messages, they commit to view change and change round from r to $r + 1$ and begin the corresponding propose phase.

The view change sub-protocol should ensure the following two properties:

- **Consistency:** If a player $\mathcal{P}_i \in H$ has committed to view-change, then any other player $\mathcal{P}_j \in H$ should also eventually commit to view-change (should not reach agreement in r).
- **Robustness:** The set T cannot launch a view-change if the leader is honest $\mathcal{P}_l \in H$.

We show in the following Claim 2 that the view-change protocol of pRFT satisfies both of these properties. We defer the proof to Appendix E.

Claim 2. *The* view-change *sub-protocol of pRFT satisfies both* Consistency *and* Robustness*.*

5.3 Discussion

The protocol leverages Proof-of-Fraud (PoF) to penalize rational players on the following π_{ds}. In addition, similar to Gilad et al. (2017) pRFT uses *tentative* and *final* consensus. We discuss (i) how PoF is realized in pRFT, (ii) the advantage of tentative and final consensus, and (iii) message complexity of pRFT.

5.3.1 Accountability and Proof-of-Fraud (PoF)

pRFT implements penalty mechanism via Proof-of-Fraud. Each player that is a part of the consensus committee $\mathcal{P}$ deposits some amount L as collateral. This collateral is locked unless some specified q number of blocks are mined. If there is some malicious behaviour by player $\mathcal{P}_i$, and a PoF exists against them, this PoF can be used as an input to the transaction to burn the collateral L of the player $\mathcal{P}_i$. Due to space constraints, we formally present the PoF construction in Appendix G. The property where deviation of more than t_0 (for some t_0) players is captured along with the identities of deviating players is called accountability and some existing consensus protocols provide accountability Civit et al. (2021); Ranchal-Pedrosa and Gramoli (2022, 2020) of deviating players. Motivating from (Civit et al., 2021, Definition 1) we define Accountability as follows:

Definition 6 (Accountability)**.** If two honest parties output different values, then eventually all honest parties reach a state s_j and receive/construct some Proof-of-Fraud (PoF) $\pi \in \{0,1\}^*$ such that $\exists$ verification algorithm $V(\cdot)$ the value $V(\pi)$ outputs set of $\geq t_0 + 1$ deviating (guilty) players.

5.3.2 Tentative and Final Consensus

After the correct execution of phases 1-3 (propose, vote and commit) of the pRFT, each player reaches a *tentative consensus* on the Block B_r. B_r reaches *final consensus* after the correct execution of step 4 if no $\mathcal{P}_i \in K$ deviates from the protocol.

Effectiveness of Tentative Consensus It is interesting to observe that tentative consensus will be finalized unless a rational player deviates from the protocol. For rational players of type $\theta = 1$ they are incentivized only in system state σ_{Fork} for which they would have to follow π_{ds}. However, as we show in the following Section 6, this deviation can be caught in phase 4 of the protocol. Due to this, any $\mathcal{P}_i \in K$ is disincentivized from deviating from the protocol, ensuring that Tentative Consensus will convert to Final Consensus.

5.3.3 Message Complexity

In normal executions, pRFT uses all-to-all broadcasts in 4 phases, leading to the message complexity n^3. Additionally, in Vote, Commit and Reveal phases, they share a set of signatures on messages from previous rounds. Therefore, the message complexity becomes $\kappa \cdot n^4$ for security parameter κ used in PKI. The message complexity and size are on par with the best protocols that provide accountability, such as Civit et al. (2021) (while pRFT tolerates a stricter adversary model compared to these solutions). We present this comparison in Table 3 (from (Civit et al., 2021, Table 1)).

6 Analysis of pRFT

We analyse the security and liveness of the protocol under partial synchronous network and threat model $\mathcal{M} = \langle(\mathcal{P}, T, K), \theta = 1, \lceil \frac{n}{4} \rceil - 1\rangle$. Therefore in the worst case, $|T| = t_0$ and $n = 4t_0 + 1$. Further, $k + t < \frac{n}{2}$. First we show that irrespective of strategy followed by $\mathcal{P}_i \in T$ if remaining $\mathcal{P} \setminus T$ do not deviate from the protocol, agreement is reached on one value (in periods of synchrony) when leader is non-deviating. In a period of asynchrony, view-change (due to timeout) happens.

Claim 3. *In any round r of pRFT such that leader $\mathcal{P}_l \notin T$, irrespective of the strategy of the set T, if remaining $\mathcal{P} \setminus T$ play $\pi \neq \pi_{fork}$ then exactly one of the following holds:*

- *If network is synchronous*[12] *and $\mathcal{P} \setminus T$ play honestly, agreement is reached on a single block B_l*
- *If network is asynchronous*[13] *or some $\mathcal{P}_i \in K$ follows $\pi \notin \{\pi_{fork}, \pi_0\}$, timeout triggers view-change.*

Proof. The proof follows by showing that any arbitrary network partition of $\mathcal{P}/\mathcal{T}$ does not lead to ≥ 2 disjoint subsets A, B such that $A \cup T$ and $B \cup T$ are $\geq n - t_0$. Therefore, either timeout happens due to insufficient ($< n - t_0$) messages in all partitions or agreement is reached in exactly one partition. The complete proof is omitted due to space constraints. □

To prove that pRFT realizes (t, k)-robust RC, we first show through Lemma 4 that any $\mathcal{P}_i \in K$ is disincentivized from deviating from the protocol i.e. following any $\pi \neq \pi_0$. We finally conclude in Theorem 5 that pRFT is a (t, k)-robust RC protocol under the threat model $\mathcal{M}$.

Lemma 4. *For any $\mathcal{P}_i \in K$ under threat model $\mathcal{M} = \langle(\mathcal{P}, K, T), \theta = 1, \lceil \frac{n}{4} \rceil - 1\rangle$ and protocol pRFT, following the protocol honestly (i.e. strategy π_0) is dominant strategy incentive compatible (DSIC) for $|K| + |T| < \frac{n}{2}$ and $t < t_0$. That is, $U_i(\pi_0, 1) \geq U_i(\pi, 1) \quad \forall\, \pi,\ \forall\, \mathcal{P}_i \in K$.*

Due to space constraints, we defer the proof to Appendix F. From Lemma 4, we conclude that any rational player of type $\theta = 1$ will follow pRFT honestly under the threat model $\mathcal{M}$ as described above. Our discussion from Section 5.3.2 implies that if all players in K behave rationally, then all tentative consensus will be converted to final consensus except during timeout due to network asynchrony. We now conclude in Theorem 5 that pRFT realizes (t, k)-robust RC.

Theorem 5. *pRFT is a (t, k)-robust rational consensus protocol under threat model $\mathcal{M} = \langle(\mathcal{P}, T, K), \theta = 1, \lceil \frac{n}{4} \rceil - 1\rangle$ for $|K| + |T| < \frac{n}{2}$ under synchronous and partial-synchronous networks.*

Proof. From Lemma 4 we know all K follow π_0. All H follow π_0 by definition of Honest players. Therefore, only deviating players are from the set T. From Claim 3 we know that in such a scenario, one of two things happen – view-change (if the network is in a period of asynchrony), or agreement (if all messages from a particular phase has been delivered. In synchronous settings, agreement is trivially satisfied due to Claim 3. In partially-synchronous setting, all messages from a round are eventually delivered before the next GST. Therefore, some block B_r (and therefore all blocks before B_r) are finalized during the period of synchrony. Since $n - t_0$ are following π_0 (and not trying to cause censorship attack), pRFT is a strongly (t, k)-robust RC protocol. □

[12] note that this period of synchrony can happen either if network is synchronous or GST event has already happened in partially-synchronous network

[13] happens in partially-synchronous networks before GST event

To summarize, we have shown through Lemma 4 that following honest strategy is DSIC for all players $\mathcal{P}_i \in K$ in partially-synchronous network. Together with Claim 3 this means under synchronous and partially synchronous networks, pRFT is strongly (t, k)-robust (Theorem 5).

7 Conclusion & Future Work

We analyzed the problem of RC under partially-synchronous network settings. We first relaxed the RC threat model discussed by Ranchal-Pedrosa and Gramoli (2022) by incorporating repeated consensus rounds and discounted utility in each round. In addition, we model types of rational players according to different types $\theta = 0, 1, 2, 3$. We showed the impossibility of achieving consensus for $\frac{n}{3} < k + t < \frac{n}{2}$ when players are incentivized to cause either liveness or censorship attacks. In that case, no consensus protocol is strongly (t, k)robust. We showed the existence of an insecure Nash Equilibrium in the consensus protocol (TRAP) explained by Ranchal-Pedrosa and Gramoli (2022). This insecure equilibrium point is preferred over the equilibrium point discussed in TRAP (Section 4.3).

We proposed pRFT, a protocol to achieve Ratoinal consensus for $k + t < \frac{n}{2}$ and $t_0 < \frac{n}{4}$. We proved pRFT security under partially synchronous network and is strong (t, k)-robust RC protocol. pRFT also achieves message complexity $O(n^3)$ and message size equal to $O(\kappa \cdot n^4)$. Through this work, we extend the understanding of the RC protocol that does not rely on baiting to achieve consensus.

Future Work. We think closing the gap between the impossibilities and guarantees by pRFT is an interesting direction for future work. In addition, improvements on pRFT via use of succinct knowledge f arguments Chen et al. (2022); Ephraim et al. (2020) is left for future work.

References

Ittai Abraham, Danny Dolev, Ivan Geffner, and Joseph Y. Halpern. 2019. Implementing Mediators with Asynchronous Cheap Talk. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing* (Toronto ON, Canada) *(PODC '19)*. Association for Computing Machinery, New York, NY, USA, 501–510. https://doi.org/10.1145/3293611.3331623

Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. 2006. Distributed Computing Meets Game Theory: Robust Mechanisms for Rational Secret Sharing and Multiparty Computation. In *Proceedings of the Twenty-Fifth Annual ACM Symposium on Principles of Distributed Computing* (Denver, Colorado, USA) *(PODC '06)*. Association for Computing Machinery, New York, NY, USA, 53–62. https://doi.org/10.1145/1146381.1146393

Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Maofan Yin. 2020. Sync HotStuff: Simple and Practical Synchronous State Machine Replication. In *2020 IEEE Symposium on Security and Privacy (SP)*. 106–118. https://doi.org/10.1109/SP40000.2020.00044

Amitanand S. Aiyer, Lorenzo Alvisi, Allen Clement, Mike Dahlin, Jean-Philippe Martin, and Carl Porth. 2005. BAR Fault Tolerance for Cooperative Services. *SIGOPS Oper. Syst. Rev.* 39, 5 (oct 2005), 45–58. https://doi.org/10.1145/1095809.1095816

Yackolley Amoussou-Guenou, Bruno Biais, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. 2020. Rational vs Byzantine Players in Consensus-Based Blockchains. In *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (Auckland, New Zealand) *(AAMAS '20)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 43–51.

Gilad Asharov, Ran Canetti, and Carmit Hazay. 2011. Towards a Game Theoretic View of Secure Computation. In *Advances in Cryptology – EUROCRYPT 2011*, Kenneth G. Paterson (Ed.). Springer Berlin Heidelberg, Berlin, Heidelberg, 426–445.

Pierre-Louis Aublin, Sonia Ben Mokhtar, and Vivien Quéma. 2013. RBFT: Redundant Byzantine Fault Tolerance. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*. 297–306. https://doi.org/10.1109/ICDCS.2013.53

Christian Badertscher, Yun Lu, and Vassilis Zikas. 2021. A Rational Protocol Treatment of 51% Attacks. In *Advances in Cryptology – CRYPTO 2021*, Tal Malkin and Chris Peikert (Eds.). Springer International Publishing, Cham, 3–32.

Erica Blum, Jonathan Katz, and Julian Loss. 2021. Tardigrade: An atomic broadcast protocol for arbitrary network conditions. In *Advances in Cryptology–ASIACRYPT 2021: 27th International Conference on the Theory and Application of Cryptology and Information Security, Singapore, December 6–10, 2021, Proceedings, Part II 27*. Springer, 547–572.

Gabriel Bracha. 1987. Asynchronous Byzantine agreement protocols. *Information and Computation* 75, 2 (1987), 130–143. https://doi.org/10.1016/0890-5401(87)90054-X

Christian Cachin, Klaus Kursawe, Frank Petzold, and Victor Shoup. 2001. Secure and Efficient Asynchronous Broadcast Protocols. https://doi.org/10.1007/3-540-44647-8_31

Christian Cachin, Klaus Kursawe, and Victor Shoup. 2000. Random Oracles in Constantipole: Practical Asynchronous Byzantine Agreement Using Cryptography. In *Proceedings of the Nineteenth Annual ACM Symposium on Principles of Distributed Computing (PODC)* (Portland, Oregon, USA) *(PODC '00)*. Association for Computing Machinery, New York, NY, USA, 123–132. https://doi.org/10.1145/343477.343531

Miguel Castro and Barbara Liskov. 1999. Practical Byzantine Fault Tolerance. In *Proceedings of the Third Symposium on Operating Systems Design and Implementation* (New Orleans, Louisiana, USA) *(OSDI '99)*. USENIX Association, USA, 173–186.

Megan Chen, Alessandro Chiesa, and Nicholas Spooner. 2022. On Succinct Non-Interactive Arguments In Relativized Worlds. In *Advances in Cryptology – EUROCRYPT 2022: 41st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Trondheim, Norway, May 30 – June 3, 2022, Proceedings, Part II* (Trondheim, Norway). Springer-Verlag, Berlin, Heidelberg, 336–366. https://doi.org/10.1007/978-3-031-07085-3_12

Pierre Civit, Seth Gilbert, and Vincent Gramoli. 2021. Polygraph: Accountable Byzantine Agreement. In *2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. 403–413. https://doi.org/10.1109/ICDCS51616.2021.00046

Flaviu Cristian, Houtan Aghili, H. Raymond Strong, and Danny Dolev. 1995. ATOMIC BROADCAST: FROM SIMPLE MESSAGE DIFFUSION TO BYZANTINE AGREEMENT. *Twenty-Fifth International Symposium on Fault-Tolerant Computing, 1995, ' Highlights from Twenty-Five Years'*. (1995), 431–.

D. Dolev and H. R. Strong. 1983. Authenticated Algorithms for Byzantine Agreement. *SIAM J. Comput.* 12, 4 (nov 1983), 656–666. https://doi.org/10.1137/0212045

Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. 1988. Consensus in the Presence of Partial Synchrony. *J. ACM* 35, 2 (apr 1988), 288–323. https://doi.org/10.1145/42282.42283

Naomi Ephraim, Cody Freitag, Ilan Komargodski, and Rafael Pass. 2020. SPARKs: Succinct Parallelizable Arguments of Knowledge. In *Advances in Cryptology – EUROCRYPT 2020*, Anne Canteaut and Yuval Ishai (Eds.). Springer International Publishing, Cham, 707–737.

Michael J. Fischer, Nancy A. Lynch, and Michael S. Paterson. 1985. Impossibility of Distributed Consensus with One Faulty Process. *J. ACM* 32, 2 (apr 1985), 374–382. https://doi.org/10.1145/3149.214121

Chaya Ganesh, Bhavana Kanukurthi, and Girisha Shankar. 2022. Secure Auctions in the Presence of Rational Adversaries. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 1173–1186. https://doi.org/10.1145/3548606.3560706

Yingzi Gao, Yuan Lu, Zhenliang Lu, Qiang Tang, Jing Xu, and Zhenfeng Zhang. 2022. Dumbo-NG: Fast Asynchronous BFT Consensus with Throughput-Oblivious Latency. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 1187–1201. https://doi.org/10.1145/3548606.3559379

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

Juan Garay, Aggelos Kiayias, and Nikos Leonardos. 2015b. The Bitcoin Backbone Protocol: Analysis and Applications. *Advances in Cryptology-EUROCRYPT 2015* (2015), 281–310.

Juan A. Garay, Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. 2013. Rational Protocol Design: Cryptography against Incentive-Driven Adversaries. *2013 IEEE 54th Annual Symposium on Foundations of Computer Science* (2013), 648–657.

Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems Principles* (Shanghai, China) *(SOSP '17)*. Association for Computing Machinery, New York, NY, USA, 51–68. https://doi.org/10.1145/3132747.3132757

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

Kostis Karantias, Aggelos Kiayias, and Dionysis Zindros. 2020. Proof-of-Burn. In *Financial Cryptography and Data Security: 24th International Conference, FC 2020* (Kota Kinabalu, Malaysia). Springer-Verlag, Berlin, Heidelberg, 523–540. https://doi.org/10.1007/978-3-030-51280-4_28

Jonathan Katz and Chiu-Yuen Koo. 2006. On expected constant-round protocols for Byzantine agreement. In *Annual International Cryptology Conference*. Springer, 445–462.

Leslie Lamport. 1998. The Part-Time Parliament. *ACM Trans. Comput. Syst.* 16, 2 (may 1998), 133–169. https://doi.org/10.1145/279227.279229

Leslie Lamport. 2001. Paxos Made Simple. *ACM SIGACT News (Distributed Computing Column) 32, 4 (Whole Number 121, December 2001)* (December 2001), 51–58. https://www.microsoft.com/en-us/research/publication/paxos-made-simple/

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

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

Yuan Lu, Zhenliang Lu, and Qiang Tang. 2022. Bolt-Dumbo Transformer: Asynchronous Consensus As Fast As the Pipelined BFT. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 2159–2173. https://doi.org/10.1145/3548606.3559346

Dahlia Malkhi, Kartik Nayak, and Ling Ren. 2019. Flexible Byzantine Fault Tolerance. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security* (London, United Kingdom) *(CCS '19)*. Association for Computing Machinery, New York, NY, USA, 1041–1053. https://doi.org/10.1145/3319535.3354225

Andrew Miller, Yu Xia, Kyle Croman, Elaine Shi, and Dawn Song. 2016. The Honey Badger of BFT Protocols. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (Vienna, Austria) *(CCS '16)*. Association for Computing Machinery, New York, NY, USA, 31–42. https://doi.org/10.1145/2976749.2978399

Satoshi Nakamoto. 2009. Bitcoin : A Peer-to-Peer Electronic Cash System.

Diego Ongaro and John Ousterhout. 2014. In Search of an Understandable Consensus Algorithm. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference* (Philadelphia, PA) *(USENIX ATC'14)*. USENIX Association, USA, 305–320.

M. Pease, R. Shostak, and L. Lamport. 1980. Reaching Agreement in the Presence of Faults. *J. ACM* 27, 2 (apr 1980), 228–234. https://doi.org/10.1145/322186.322188

Alejandro Ranchal-Pedrosa and Vincent Gramoli. 2020. ZLB: A blockchain to tolerate colluding majorities. *arXiv preprint arXiv:2007.10541* (2020).

Alejandro Ranchal-Pedrosa and Vincent Gramoli. 2022. TRAP: The Bait of Rational Players to Solve Byzantine Consensus. In *Proceedings of the 2022 ACM on Asia Conference on Computer and Communications Security* (Nagasaki, Japan) *(ASIA CCS '22)*. Association for Computing Machinery, New York, NY, USA, 168–181. https://doi.org/10.1145/3488932.3517386

Thomas C. Schelling. 1963. *The strategy of Conflict*. Oxford University Press.

Alexander Spiegelman, Neil Giridharan, Alberto Sonnino, and Lefteris Kokoris-Kogias. 2022. Bullshark: DAG BFT Protocols Made Practical. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 2705–2718. https://doi.org/10.1145/3548606.3559361

Xavier Vilaça, João Leitão, Miguel Correia, and Luís Rodrigues. 2011. N-party BAR Transfer. In *Principles of Distributed Systems*, Antonio Fernàndez Anta, Giuseppe Lipari, and Matthieu Roy (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 392–408.

Xavier Vilaça, João Leitão, and Luís Rodrigues. 2011. N-Party BAR Transfer: Motivation, Definition, and Challenges. In *Proceedings of the 3rd International Workshop on Theoretical Aspects of Dynamic Distributed Systems* (Rome, Italy) *(TADDS '11)*. Association for Computing Machinery, New York, NY, USA, 18–22. https://doi.org/10.1145/2034640.2034647

Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham. 2019. HotStuff: BFT Consensus with Linearity and Responsiveness. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing* (Toronto ON, Canada) *(PODC '19)*. Association for Computing Machinery, New York, NY, USA, 347–356. https://doi.org/10.1145/3293611.3331591

Haibin Zhang and Sisi Duan. 2022. PACE: Fully Parallelizable BFT from Reproposable Byzantine Agreement. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (Los Angeles, CA, USA) *(CCS '22)*. Association for Computing Machinery, New York, NY, USA, 3151–3164. https://doi.org/10.1145/3548606.3559348

A Definitions

A.1 Byzantine Agreement

Definition 7 (Byzantine Agreement Blum et al. (2021) (BA))**.** A protocol Π is run by a set of players is $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2 \ldots, \mathcal{P}_n\}$ and each player $\mathcal{P}_i$ initially has value v_i. This protocol Π solves *byzantine agreement* t-securely if it satisfies:

- **Agreement:** All altruistic players decide on the same value.
- **Validity:** If all altruistic players have the same input value v then they agree on the same value v.
- **Termination:** All honest players eventually decide on some value and protocol Π terminates.

Byzantine Agreement (BA) and Byzantine Broadcast (BB) are equivalent problems in the domain of distributed cryptographic protocols. This means that we can use BA (and equivalently BB) as a black box to implement BB (and equivalently BA). BA/BB is possible only if $t < n/3$ in asynchronous and partially-synchronous networks and $t < n/2$ under synchronous network settings Blum et al. (2021); Lamport et al. (1982b).

A.2 Atomic Broadcast

Definition 8 (Atomic Broadcast Blum et al. (2021) (ABC))**.** A protocol Π which is executed by players $\mathcal{P} = \{\mathcal{P}_1, \mathcal{P}_2 \ldots, \mathcal{P}_n\}$ who are provided with transactions and maintain a list (chain) of Blocks implements *Atomic Broadcast* t-securely if it satisfies the following conditions:

- t-completeness i.e. if $\leq t$ players are corrupted, then $\forall j > 0$, each honest player outputs block in *iteration* j.
- t-consistency i.e. if $\leq t$ players are corrupted, then $\forall j > 0$ if one honest player outputs Block B in iteration j then all honest players output B.
- t-liveness i.e. if $\leq t$ players are corrupted, then if all honest players have transaction tx as input, then all honest players eventually output a block containing tx.

Atomic Broadcast is the abstraction that is realized by Blockchain protocols. While randomized and probabilistic consensus protocols such as Proof-of-Work (PoW) Nakamoto (2009); Garay et al. (2015b), Proof-of-Stake (PoS) Gilad et al. (2017) among others, solve ABC with high probability, protocols such as pBFT Castro and Liskov (1999), Honeybadger Miller et al. (2016) etc. deterministically solve ABC.

A.3 Flavours of Synchrony

Consensus and Broadcast problems are studied for a set of players connected through a network. This network is either assumed as *synchronous, asynchronous* or *partially-synchronous*.

- **Synchronous:** A network is fully-synchronous (or just *synchronous*) if for any message sent from sender S to receiver R reaches within some delay which has an upper bound Δ_{sync} known to the protocol in advance. Therefore, protocols that work under a synchronous network can be parameterized using this delay parameter Δ_{sync}.
- **Asynchronous:** A network is asynchronous if, for any message from sender S to receiver R, the delay has no upper bound but is a finite value. This would mean each message is guaranteed to be delivered, but there exists no upper bound on the delay.
- **Partially-Synchronous:** Partial-synchronous networks are intermediaries between synchronous networks – which are difficult to realize and asynchronous networks – under which designing protocols is challenging. Partially-Synchronous networks were first discussed by Dwork et al. Dwork et al. (1988) and are defined as a network having some finite delay Δ_{ps} set by the adversary and is not known to the protocol. Therefore, a protocol satisfying consensus under partial synchrony cannot be parameterized using Δ_{ps} and should satisfy any $\Delta_{ps} \in \mathbb{R}_{>0}$. However, the protocols can use the existence of a finite upper bound to network delay to realize functionalities that were difficult (or impossible) in asynchronous settings.

Consensus through a *deterministic protocol* is impossible under asynchronous network settings in the presence of even a single faulty party Fischer et al. (1985). Consensus under a synchronous network is possible using the deterministic

protocol as long as the majority is honest, i.e. byzantine players $< \frac{n}{2}$. Under partial synchrony, consensus is possible if byzantine players are $< \frac{n}{3}$. We discuss RC in Partially-Synchronous network settings and aim to propose a more relaxed adversary model with t byzantine and k rational players and exploit rationality of k players to achieve consensus in $k + t < \frac{n}{2}$.

B Proof for Theorem 1

Theorem 6 (Rational Consensus under $\theta = 3$)**.** *Under the threat model $\langle(\mathcal{P}, T, K), \theta = 3, t_0\rangle$ no rational consensus protocol is (t, k)-robust when $\lceil \frac{n}{3} \rceil \leq k + t \leq \lceil \frac{n}{2} \rceil - 1$.*

Proof. Consider a protocol Π belonging to the set of protocols $\mathcal{C}_3$ that achieve consensus under threat model $\mathcal{M} = \langle(\mathcal{P}, T, K), \theta = 3, t_0\rangle$. Let $n = |\mathcal{P}|$ therefore, $|T| + |K| < \frac{n}{2}$. In such a case, consider the threshold of messages required for agreement by the protocol to be τ. From Claim 1 we have $\tau > \lfloor \frac{n+t_0}{2} \rfloor \geq \lceil \frac{n}{2} \rceil$ (since $t_0 \geq 1$). This means that for consensus, the message/signature of at least one player in $K \cup T$ is required. In this case, if each player $\mathcal{P}_l \in K \cup T$ follows strategy π_{abs} – not sending any messages, then consensus is not reached. Since abstaining from sending messages in a round is indistinguishable from message delays due to partially-synchronous networks, π_{abs} cannot be distinguished from π_0 under partially-synchronous network. Therefore, $D(\pi_{abs}, \sigma) = 0$. The utility for following π_{abs} for each rational player is

$$\begin{aligned} U_i(\pi_{abs}, \theta = 3) =& \sum_{r=0}^{\infty} \delta^r u_i(\pi_{abs}, 3, r) \\ =& E_{\sigma \sim S}[f(\sigma, 3)] - 0 \\ =& \alpha > 0 = U_i(\pi_0, \theta = 3) \end{aligned}$$

Therefore, each player in set K is incentivized to play π_{abs} over following protocol which compromises (t, k)-eventual liveness property and therefore any such arbitrary Π is not a (t, k)-robust Rational Consensus protocol. □

C Proof for Theorem 2

Theorem 7 (Rational Consensus under $\theta = 2$)**.** *Under threat model $\langle(\mathcal{P}, T, K), \theta = 2, t_0\rangle$ no rational consensus protocol is strongly (t, k)-robust when $\lceil \frac{n}{3} \rceil \leq t + k \leq \lceil \frac{n}{2} \rceil - 1$.*

Proof. The proof follows by showing a strategy followed by $\mathcal{P}_i \in K \cup T$ that is incentivized for rational players and following this strategy, for any protocol Π it is impossible to achieve strongly (t, k)-robust rational consensus. In Step 1, we describe the strategy, in Step 2, we show that rational players are incentivized to follow this strategy and in Step 3 we show that for any protocol Π it is impossible to achieve rational consensus under this strategy.

Step 1 (Strategy π_{pc}): From Theorem 1 we have that for any protocol Π with $\tau \geq \lfloor \frac{n+t_0}{2} \rfloor + 1$ in any round r, the collusion $K \cup T$ can cause disagreement by following π_{abs}. Let us consider a transaction tx_h which is input to all honest players by round r_0. The strategy which $\mathcal{P}_i \in K \cup T$ follows for round $r \geq r_0$ is:

- If leader in round r is $\mathcal{P}_l \notin K \cup T$ then follow π_{abs}.
- If leader in round r is $\mathcal{P}_l \in K \cup T$ then propose Block with transaction set $\overline{tx}$ such that $tx_h \notin \overline{tx}$.

We abbreviate this strategy as π_{pc} (**p**artial-**c**ensorship).

Step 2 (Incentive Compatibility): We now show that following π_{pc} is incentivized for rational players $\mathcal{P}_i \in K$. We first make a simple observation that in from round r_0 to $r_0 + n - 1$, in expectation there will be $k + t$ blocks mined (when the leader is $\mathcal{P}_l \in K \cup T$). Therefore, the protocol achieves (t, k)-eventual liveness. In addition, since there are no conflicting values proposed in any round, no disagreement is reached. The rational players are of type $\theta = 2$ which means their utility from round r_0 onwards is given by

$$U_i(\pi, \theta = 2) = \sum_{r=r_0}^{\infty} \delta^r \left(\mathbb{E}[\alpha f(\sigma, 2)] - L \cdot D(\pi) \right)$$

Since system will not be in state σ_{NP}, the payoff from $\mathbb{E}[\alpha_2 f_2] > 0$ if probability $Pr(\sigma = \sigma_{CP}) > 0$ (according to our strategy $Pr(\sigma = \sigma_{CP}) = 1$). In addition, since there are no duplicate messages signed, and players do not crash forever, following π_{pc} is indistinguishable from following π_0 which means $D(\pi_{pc}) = D(\pi_0) = 0$. This means $\forall \mathcal{P}_i \in K \cup T$

$$U_i(\pi_{pc}, \theta = 2) > U_i(\pi_0, \theta = 2)$$

The set $K \cup T$ is therefore incentivized to deviate from the honest protocol π_0 to follow π_{pc} for any protocol Π.

Step 3 (No strongly (t,k)-robust Rational Consensus: We now argue that if $K \cup T$ follows π_{pc} for any consensus protocol then it is impossible to achieve strongly (t,k)-robust rational consensus for any $\frac{n}{3} \leq k + t < \frac{n}{2}$. Consider any round $r \geq r_0$. In this round, if leader $\mathcal{P}_l \in K \cup T$ the leader selectively includes transactions in transaction set $\overline{tx}$ such that $tx_h \notin \overline{tx}$. If $\mathcal{P}_l \notin K \cup T$ the coalition causes view change without agreement on a block. Thus, any block confirmed (and thus included) doesnot contain transactoin tx_h although the transaction is input to all honest players at round r_0. This violates the (t,k)-censorship resistance and therefore for any arbitrary protocol[14] Π achieving strongly (t,k)-robust rational consensus is impossible. □

D Proof for Theorem 3

Theorem 8 (Baiting based Rational Consensus under $\theta = 1$)**.** *Consider any baiting-based rational consensus protocol Π the threat model $\mathcal{M} = \langle(\mathcal{P}, K, T), \theta = 1, t_0\rangle$. The set of rational players following π_{fork} is a Nash-equilibrium strategy for $|K| > 2 + t_0 - t$. Under this strategy, the protocol Π fails to achieve (t,k)-robust rational consensus for $t_0 = \lceil \frac{n}{3} \rceil - 1$.*

Proof. To prove this, we show that the payoff for a rational player $\mathcal{P}_i$ on following π_{bait} is lesser than the payoff if they follow π_{Fork} (which is the strategy followed by the collusion). The net gain in payoff for the collusion $K \cup T$ is G if the system ends up in state σ_{Fork} which is distributed among the rational players. If the player $\mathcal{P}_i \in K$ follows π_{bait} then she is not a part of the collusion and will get 0 payoff if the system ends up in σ_{Fork}.

The payoff for $\mathcal{P}_l$ on following π_{Fork} along with the rest of the coalition $K \cup T$ is therefore

$$U_i(\pi_{Fork}, 3) = \frac{G}{k}$$

Consider for $|K| > 2 + t_0 - t$ i.e. $k + t > 2 + t_0$. If $\mathcal{P}_l$ deviates from the coalition $K \cup T$ to follow π_{bait}. The system can end up in two possible states. σ_{Fork} is if the rest of the collusion $T \cup K/\{\mathcal{P}_l\}$ is able to create a fork/disagreement despite $\mathcal{P}_l$ following π_{bait}. In this case, the utility for $\mathcal{P}_l$ is 0 (since they were not a part of the collusion). Second is if the Proof-of-Fraud submitted by $\mathcal{P}_l$ is accepted and the protocol functions normally σ_0. The payoff in this case is

$$U_i(\pi_{bait}, 3) = R \cdot \ Pr(\sigma = \sigma_0) + 0 \cdot \ Pr(\sigma = \sigma_{Fork})$$

However, if $k + t > 2 + t_0$ then we can achieve disagreement even if $1 + t_0$ players follow π_{Fork} (since $t_0 + 1 = \lceil \frac{n}{3} \rceil$). This can be under the partition of $\mathcal{P}/(T \cup K/\{\mathcal{P}_i\})$ into two disjoint sets A, B such that $|A|$ and $|B|$ are such that $|A| + k + t \geq \tau$ (and similarly $|B|$). For forking, we have to ensure each partition has at $n - t_0$ messages. If m players from the collusion deviates to follow π_{bait} the condition for system to **not** end in $\sigma = \sigma_{Fork}$ is

$$|A| + (k - m) + t < \tau < n - t_0$$
$$m > |A| + k + t - n + t_0$$

We have $n - |B| = |A| + k + t$ and $|B| = \frac{n-t-k}{2}$. Therefore,

$$m > t_0 + \frac{k + t - n}{2}$$

For $k > 3 > (3t_0 + 1) - 2t_0 - t + 2 = n - 2t_0 - t + 2$, we have from algebraic reordering the RHS of inequality > 1. Thus, any unilateral deviation ($m = 1$) is not sufficient to avoid σ_{Fork}. Therefore, $Pr(\sigma = \sigma_{Fork}) = 1$ and $Pr(\sigma = \sigma_0) = 0$. The utility is therefore 0 which gives us $U_i(\pi_{Fork}, 3) > U_i(\pi_{bait}, 3)$. Following π_{abs} will also lead to $\mathcal{P}_i$, not part of $K \cup T$ and therefore the payoff is 0. Following π_0 leads to payoff 0 for any system state. Thus, following π_{Fork} is Nash Equilibrium strategy for $\mathcal{P}_l \in K$. □

[14] we did not assume any property about the protocol

E Proof for Claim 2

Claim 4. *The* view-change *sub-protocol of* pRFT *satisfies both* Consistency *and* Robustness*.*

Proof. We prove Consistency through contradiction. Consider $\mathcal{P}_1 \in A$, $\mathcal{P}_2 \in B$ such that $A, B \subset H, A \cap B = \emptyset$. Assume consistency does not hold, i.e. $\mathcal{P}_1$ commits to view-change and broadcasts $CommitView$ message (ref. Table 2b). Since communication channels are reliable and messages are not dropped, the only way to disrupt consistency is if $\mathcal{P}_2$ reaches agreement before this $CommitView$ message from $\mathcal{P}_1$ reaches $\mathcal{P}_2$. For this, we require $|B| + k + t \geq n - t_0$. However, for a valid $CommitView$ we require $|A| + k + t \geq n - t_0$. Adding them up, $|A| + |B| + 2(k + t) > 2n - 2t_0$. This gives us $n + k + t > |A| + |B| + 2(k + t) \geq 2n - 2t_0$ and on rearrangement $k + t + 2t_0 > n$ which is a contradiction for the considered threat model $\mathcal{M} = \langle(\mathcal{P}, T, K), \theta = 1, \frac{n}{4}\rangle$. Therefore, consistency is satisfied.

For robustness, consider in a round with honest leader $\mathcal{P}_l \in H$. In this case, if $\mathcal{P}_i \in T$ broadcast $ViewChange$ message, and abstain from participation, still the protocol is able to gather $\geq n - t_0$ message. This is because (1) $t \leq t_0$ and (2) rational players are of type $\theta = 1$ due to which they are disincentivized from causing liveness attack. Hence, not enough ($\geq n - t_0$) view change messages are gathered and T cannot cause view-change by themselves. The protocol therefore satisfies Robustness property. □

F Proof for Lemma 4

Lemma 9. *For any $\mathcal{P}_i \in K$ under threat model $\mathcal{M} = \langle(\mathcal{P}, K, T), \theta = 1, \lceil\frac{n}{4}\rceil - 1\rangle$ and protocol* pRFT*, following the protocol honestly (i.e. strategy π_0) is dominant strategy incentive compatible (DSIC) for $|K| + |T| < \frac{n}{2}$ and $t < t_0$.*

$$U_i(\pi_0, 1) \geq U_i(\pi, 1) \quad \forall\ \pi,\ \forall\ \mathcal{P}_i \in K$$

Proof. Consider any arbitrary rational player $\mathcal{P}_i \in K$. We show that by playing π_{fork}, $\mathcal{P}_i$ either: (1) get caught in the PoF, (2) cause view-change or (3) cause agreement on a single value. Consider π_{fork} is played by $\mathcal{P}_i$ and two honest players $\mathcal{P}_a, \mathcal{P}_b \in H$ receive conflicting signatures on value h_a, h_b (such that $h_a \neq h_b$). First, consider the case when network is synchronous (messages are reaching on time). In this case, signature on h_a reaches to $\mathcal{P}_b$ and similarly signature on h_b reaches $\mathcal{P}_a$. Either number of double signatures are $\leq t_0$ in which case, agreement is satisfied due to Claim 3. If number of double signature is $> t_0$ then PoD is constructed by either (or both) $\mathcal{P}_a$ or (and) $\mathcal{P}_b$ and $\mathcal{P}_i$ suffers penalty (with some non-zero probability). The payoff in this case is $u_i(\pi_{fork}, 1, r) = -L \cdot D(\pi_{fork}, \sigma) < 0$. Consider if the network is partially-synchronous and the network partition of honest nodes is such that A, B are two partitions and $\mathcal{P}_a \in A$ and $\mathcal{P}_b \in B$. If either partition is small enough that $k + t + |A| < n - t_0$ or $k + t + |B| < n - t_0$, then agreement is not reached in that partition. In this case, either agreement is not reached for both $\mathcal{P}_a$ and $\mathcal{P}_b$ or agreement is reached for exactly one of $\mathcal{P}_a$ or $\mathcal{P}_b$ (therefore on one same value). Let the probability of no agreement be q_d and agreement on exactly one value be q_a. The utility is therefore, $u_i(\pi_{fork}, 1, r) = q_d f(\sigma_{NP}, 1) = -q_d\alpha \leq 0$. Notice that it cannot be that agreement is reached in both partitions, as in that case, $|A| + k + t > n - t_0$ and $|B| + k + t > n - t_0$. As $|A| + |B| < |H|$, Adding them up, we have $2n - 2t_0 \leq n - (k + t) + 2(k + t) \Rightarrow n < k + t + 2t_0$.

However, according to our threat model $\mathcal{M}$, $k + t < \frac{n}{2}$ and $t_0 < \frac{n}{4}$. Therefore, $k + t + 2t_0 < n$. Therefore, such a case is not possible in our threat model. Therefore, if the network in (or during) that round is synchronous, $u_i(\pi_{fork}, 1, r) < 0$ and under partially-synchronous network $u_i(\pi_{fork}, 1, r) \leq 0$. Therefore, the expected utility for the round is $\forall i \in [n], r \in \mathbb{R}$

$$\mathbb{E}_{\sigma\sim S}[u_i(\pi_{fork}, 1, r)] \leq 0 \implies U_i(\pi_{fork}, 1) \leq 0 = U_i(\pi_0, 1)$$

Following any other strategy will lead to at most t_0 double signatures (from byzantine players) and from Claim 3, the protocol will either reach an agreement or view-change (through timeout or duplicate values proposed by the adversarial leader). Additionally, since rational players are of type $\theta = 1$, they will not try to cause censorship of transactions. Therefore, following $\pi \notin \{\pi_0, \pi_{fork}\}$ for $\mathcal{P}_i \in K$ gives

$$U_i(\pi, 1) \leq 0 = U_i(\pi_0, 1)$$

Hence, following π_0 gives more payoff than any other strategy. Thus following pRFT is DSIC for any $\mathcal{P}_i \in K$. □

G Construct Proof Procedure

We elaborate the construct proof procedure invoked in the Reveal phase of the pRFT protocol (Figure 1) through Figure 4.

ConstructProof(M, t_0)

```
1:  D := ∅
2:  for i ∈ [n], j ∈ [n]/{i} do
3:      for k ∈ [n] do
4:          if M(i,k) ≠ M(j,k) then
5:              D ← D ∪ {(M(i,k), M(j,k)} and goto 9
6:          end if
7:      end for
8:      if |D| ≥ t_0 + 1 then
9:          return D
10:     end if
11: end for
12: return D
```

Figure 4: Construction of Proof-of-Fraud (PoF)