Termination are guaranteed. Our findings are as follows: when $f \geq \nu$, invalid blocks are accepted, so that validity is not satisfied; when $f < \nu$, while there exists an equilibrium in which validity and termination are satisfied, there also exists an equilibrium in which blocks are never accepted, so that termination is not satisfied. This points to a tension between validity (which requires that the ν threshold be large enough) and termination (which can be threatened when the ν threshold is high.).

Related work. Rational participants have been considered in various works in distributed computing, e.g. [1–3, 5, 6, 20, 22, 23, 26]. [1] shows some advantages of combining game theory and distributed computing and presents some challenges. As for Byzantine consensus, the utilities of rational players take into account whether or not a decision is reached. For the problem of Byzantine agreement for instance, Groce *et al.* [22] consider an environment with rational and honest participants where they provide protocols that tolerate rational adversaries and proved lower bounds. In [23], Halpern and Vilaça prove that in a full rational setting, if participants can fail by crashing, then there is no *ex post* Nash equilibrium solving the *fair* consensus problem (where fair means that the input of every agent is decided with equal probability), even with only one crash. They also present a protocol satisfying fair consensus under some assumptions over the failures patterns. [5] proposes building blocks for distributed algorithms and proposes protocols solving consensus and renaming. For leader election, [3] shows that in systems with only rational players, under certain conditions, it is possible to obtain a k-resilient equilibrium (resistant to a coalition of up to k participants). Here, we consider the costs of actions (Sending, Checking) in the players' utilities, and not only if a decision is reached, contrary to the previous works. We also consider an environment composed of a combination of Byzantine and rational players.

Similar to our study, in [26], Lysyanskaya and Triandopoulos consider rational and Byzantine participants, while studying the problems of secret sharing and multi-party computation; the latter subsumes classical consensus protocols. [26] proposes an incentive compatible protocol resistant to a coalition of up to f faulty players, where the utilities consider if a decision is reached or not, and on the value of decision, they also analyzed the case where Byzantine utilities may be unknown. Concurrently to [26], [2] proposes an incentive-compatible protocol for secret sharing with rational participants where some utilities can be unknown. Contrarily to previous studies, we consider the costs of actions (Sending, Checking) in the players utilities; these costs make the study more realistic and the analysis more complex. Our approach also differs from the previous works by the following: when previous works provide an incentive compatible protocol for rational participants, we take a protocol solving the consensus problem in the classical honest vs Byzantine setting (no rational), and then study the rational behaviors with that protocol. In particular, in this work we do not only show that there exist a *good* equilibrium, but we also exhibit other different equilibria that exists in the setting. The articles [2, 22, 26], use cooperative game theory in their analyses, where we follow the BAR model [6] and study the rational behavior in a non-cooperative setting, as in [23].

Blockchains either use proof-of-* like mechanisms (e.g. proof-of-work, proof-of-stake, proof-of-elapsed-time, etc.) or classical Byzantine consensus approaches [4, 7, 16, 17, 24]. Byzantine consensus-based blockchains have the advantage to guarantee strong consistency by running a Byzantine Fault Tolerant (BFT) protocol [13]. In order to use a BFT protocol in an open setting, recent research has been devoted to either find secure mechanisms to select committees of fixed size over time (e.g. [15, 21]) and/or to propose incentives to promote participation [4]. Most of the proposals, however, assume participants as either honest or Byzantine, failing to thoroughly explore the effect of rational participants. In this line of work, Solidus [4] is the first to consider rational participants by proposing an incentive-compatible BFT protocol for blockchains. Solidus introduces interesting incentive mechanisms; however, the work does not provide a game theoretic framework for their analysis.

While addressing a different protocol, [27] is the closest work to ours. In this protocol, multiple committees run in parallel to validate a non-intersecting set of transactions (a shard). A non-cooperative static game approach for the intra-committee protocol is taken leading to the result that rational agents can free-ride when rewards are equally shared. The main aspect of our analysis that is new and different from [27] is the following: we have a dynamic (not static) multi-round analysis of a problem in which some participants are Byzantine and some blocks can be invalid (and costly for rational players if accepted). In that context, there is a situation in which in equilibrium rational agents are pivotal, because if they do not check the block validity this will create the risk of having an invalid block accepted. It is because they are pivotal that they do not free ride. Moreover, we discuss equilibria in relation to formal consensus properties – Termination vs Validity –, which represents a novelty.

In the realm of consensus-free blockchains (e.g. Bitcoin) many works used rational arguments to prove thresholds on the fraction of honest nodes needed to guarantee security properties [18, 32]. These works establish very pessimistic thresholds while in practice Bitcoin works even if the honest majority assumption does not hold. Following this observation, [9] proposes a rational analysis of Bitcoin based on the rational protocol design framework [20]. The proposed game, with respect to ours, is at an upper level of abstraction, proposing a two-player game between the protocol designer and the adversary. The rational protocol design also models the behavior of (some of the) protocol participants as rational, and studies the general problem of secure multi-party computation. Our game models instead the behavior of protocol participants that can be rational, evolving in an environment with Byzantine participants. Moreover, our work targets consensus-based blockchain unlike [9].

With only rational players, [10] models Bitcoin as a coordination game. Similar to the work in [10], our analysis shows that the protocol in consensus-based blockchains is a coordination game. Additionally, we consider Byzantine players, and show that *Termination* can be violated when coordination failures occur. [31] uses game theory to study consensus-free *proof-of-stake* blockchains, and shows that the *Nothing at Stake* problem is mitigated because players with large stakes on the main chain prefer not to add blocks on forking branches, since it reduces the strength of the

main chain, and thus the value of their stakes. The environment considered in [31] differs from ours, since [31] does not consider consensus-based blockchains, nor Byzantine players.

2 BLOCKCHAIN CONSENSUS WITH RATIONAL PLAYERS

2.1 System Model

We consider a system composed of a finite and ordered set Π, called *committee*, of synchronous sequential processes or players, namely $\Pi = \{p_1, \ldots, p_n\}$ where process p_i is said to have index i. We assume each player is aware of its index. In the following, we refer to process/player p_i by its index, say process i. Hereafter, the words "player" and "process" are taken to have the same meaning.

Communication. We assume that each process evolves in rounds. A *round* consists of one or more phases, and each phase is divided into three sequential steps, in order: the send, the delivery and the compute step. We assume that the send step is atomically executed at the beginning of the phase and the compute step is atomically executed at the end of the phase. The phase has a fixed duration that allows collecting all the messages sent by the processes at the beginning of the phase during the delivery step. At the end of a phase, a process exits from the current phase and starts the next one. The processes communicate by sending and receiving messages through a reliable broadcast primitive. Messages are created with a digital signature, and we assume that digital signatures cannot be forged. When a process i delivers a message, it knows the process j that created the message. We assume that messages cannot be lost.

Processes Behavior. In this paper we consider a variant of the BAR model [6] where processes are either *rational* or *Byzantine*. *Rational processes* are self-interested and target to maximize their expected utility. They will deviate from a prescribed (suggested) protocol if and only if doing so increases their expected utility. Their objective function must account for their costs (e.g., sending messages) and benefits (e.g., reward of a block) for participating in a system. In line with [6], the objective of Byzantine processes is to prevent the protocol from achieving its goal, and to reduce the rational processes utility, no matter the cost. We denote by f the number of Byzantine processes in the network. We assume *symmetric Byzantines* (i.e. their behavior is perceived identically by all non Byzantine processes). That is, a message sent by a Byzantine process and received by a non-Byzantine process in a given phase is received by all non-Byzantine processes in the same phase. Note that this can be implemented by introducing an extra phase of communication.

2.2 Byzantine Consensus-based Blockchain

Consensus-based blockchains should satisfy the following consensus properties: *Termination:* every non-Byzantine process decides on a value (a block); *Agreement:* if two non-Byzantine processes decide respectively on values B and B', then $B = B'$; *Validity[12, 14]:* a decided value by any non-Byzantine process is valid, it satisfies the predefined predicate. Let us note that the above properties must hold also for systems prone to rational behaviors.

A blockchain is a growing sequence of blocks, where any new block in the blockchain can only be appended. Once a block is in the blockchain, it cannot be modified nor removed. The block at position $h \geq 0$ in the blockchain is said to be at height h, and the first block at height 0 is the initialization block. Let us call *length* of a blockchain the number of blocks the blockchain is composed of. If the length of the blockchain is $h - 1 \geq 0$, then the next block to be added to the blockchain can only be at height h. To implement the above specification in consensus-based blockchains, for each height $h > 0$ of the blockchain, a consensus instance is run inside a committee selected for the given height. In this paper, we analyze a very general protocol, inspired by [4, 7, 16, 17, 24, 28, 33], variants of PBFT [13]. In this protocol, a proposer proposes a value, i.e. a block, and the other members of the committee will check the validity of the value. If the value is valid, then they will vote for it and will announce their vote through a message to the other members. Votes are collected and if a given threshold is reached, then the value is decided, otherwise a new proposer will propose another block and the procedure restarts.

The prescribed protocol. The protocol proceeds in rounds. For sake of simplicity we consider the height h of the blockchain passed as parameter to the protocol. Algorithm 1 presents the pseudo-code of the protocol.

Algorithm 1 Prescribed Protocol for a given height h at any process i

```
1:  Initialization:
2:    vote := nil
3:    r := 0                                                    /* Current round number */
4:    decidedValue := nil

5:  Phase PROPOSE(r):
6:    Send step:
7:      if i == isProposer(r, h) then
8:        proposal ← createValidValue(h)      /* The proposer of the round generates a
          block, i.e. the value to be proposed */
9:        broadcast ⟨PROPOSE, h, r, proposal⟩
10:   Delivery step:
11:     delivery ⟨PROPOSE, h, r, v⟩ from proposer(r)   /* The process collects the proposal
        */
12:   Compute step:
13:     if isValid(v) then
14:       vote ← v    /* If the delivered proposal is valid, then the process sets a vote for it */

15: Phase VOTE(r):
16:   Send step:
17:     if vote ≠ nil then
18:       broadcast ⟨VOTE_i, h, r, vote⟩ /* If the proposal is valid, the process sends the vote
          for it to all the validators */
19:   Delivery step:
20:     delivery ⟨VOTE, h, r, v⟩     /* The process collects all the votes for the current height
        and round */
21:   Compute step:
22:     if |⟨VOTE, h, r, v⟩| ≥ v ∧ decidedValue = nil ∧ vote ≠ nil ∧ vote = v then
23:       decidedValue ← v; exit /* The valid value is decided if the threshold is reached */
24:     else
25:       vote ← nil
26:       r ← r + 1
```

For each round t a committee member is designated as the proposer for the round in a round robin fashion. The `isProposer`(t, h) function returns the id of the proposer for the current round (line 7). The function, by taking as parameter the current height, deterministically selects the proposer on the basis of the information contained in the blockchain up to h (the actual selection mechanism is out of the scope of the paper). Each round is further divided in two phases: the PROPOSE and the VOTE phase.

During the PROPOSE phase, the proposer of the round uses the function `createValidValue`(h) to generate a block. Because a valid block must include the identifier of the h^{th} block in the blockchain, the height h is passed as parameter (line 8). Once the block is created, a message broadcasting the proposal is sent (line 9). At line 10 the proposal is received through a delivery function. Each process checks if the proposal is a valid value (line 13). If so, the process sets its vote to the value (line 14).

During the VOTE phase, any process that sets its vote to the current valid proposal sends a message (of type vote) to the other members of the committee (line 18). During the delivery step, sent messages are collected by every process. During the compute step, each process verifies if a quorum of v votes for the current proposal has been reached. Let us note that v, the majority threshold is a parameter here, because it is the object of our study to establish the quorum v in presence of rational and Byzantine processes. If the quorum is reached, the process voted for the value and did not already decide for the current height, then it decides for the current proposal (line 23) and the protocol ends. If the quorum is not reached, then a new round starts (line 26).

Let us note that the protocol in an environment assuming only correct and symmetric Byzantine processes trivially implements consensus if f, the number of Byzantine processes, is such that $f < v$. If $f \geq v$, on the other hand, the Termination property is not guaranteed. The scenario for that is that Byzantine validators might vote for a different value with respect to the one voted by correct processes or a nil value. In that case, the correct process will not decide (line 22) and will move in the next round. The scenario can repeat forever.

In the following we detail the pseudo-code for a rational process shown in Algorithm 2. The rational process will try to maximize its payoff by choosing to undertake or not the actions defined in its action space (Section 2.3). We consider the choice of: (i) proposing or not a valid block, (ii) checking or not the validity of a block and (iii) sending or not the vote for a proposed block. We consider that the action of checking the validity of the block and the action of sending the message (of type vote) have a cost.

Protocol of the rational processes. Rational processes choices are explicitly represented in the pseudo-code (Algorithm 2) by dedicated variables, namely, $action^{propose}$, $action^{check}$, and $action^{send}$, defined at lines $5-7$. Each action, initialized to nil, can take values from the set $\{0, 1\}$. Those values are set by calling the functions $\sigma_i^{\text{propose}}$, σ_i^{check}, and σ_i^{send}, respectively, returning the strategy for the process i.

Strategy $\sigma_i^{\text{propose}}$ determines if the proposer i chooses to produce a valid proposal or an invalid one (lines 12-16). In both cases, the proposal is sent in broadcast (line 17).

Strategy σ_i^{check} determines if the receiving process chooses to check the validity of the proposal or not, which is a costly action. If the process chooses to check the validity (line 22), it will also update the knowledge it has about the validity of the proposal and it will pay a cost c_{check}. If otherwise, the process keeps not knowing if the proposal is valid or not ($validValue[r]$ remains set to $\perp$). Note that this value remains set to $\perp$ even if the process is the proposer. This is because we assumed, without loss of generality, that checking validity has a cost and that the only way of checking validity is by executing the `isValid`(v) function.

Algorithm 2 Pseudo-code for a given height h modeling the rational process i's behavior

```
1:  Initialization:
2:    vote := nil
3:    r := 0                                                    /* Current round number */
4:    decidedValue := nil
5:    action^propose := nil
6:    action^check := nil
7:    action^send := nil
8:    validValue[] := {⊥, ⊥, ..., ⊥}                            /* validValue[r] ∈ {⊥, 0, 1} */

9:  Phase PROPOSE(r):
10:   Send step:
11:     if i == isProposer(h, r) then
12:       action^propose ← σ_i^propose()   /* σ_i^propose ∈ {0, 1} sets the action of proposing a
          valid block or an invalid one */
13:       if action^propose == 1 then
14:         proposal ← createValidValue(h)
15:       else if action^propose == 0 then
16:         proposal ← createInvalidValue()
17:       broadcast ⟨PROPOSE, h, r, proposal⟩
18:   Delivery step:
19:     delivery ⟨PROPOSE, h, r, v⟩ from proposer(h, r)
20:   Compute step:
21:     action^check ← σ_i^check()   /* σ_i^check ∈ {0, 1} sets the action of checking or not the
          validity of the proposal */
22:     if action^check == 1 then
23:       validValue[r] ← isValid(v)   /* The execution of isValid(v) has a cost c_check */
24:     action^send ← σ_i^send(validValue)   /* σ_i^send : {⊥, 0, 1} → {0, 1} sets the action of
          sending the vote or not */
25:     if action^send == 1 then
26:       vote ← v   /* The process decides to send the vote, the proposal might be invalid */

27: Phase VOTE(r):
28:   Send step:
29:     if vote ≠ nil then
30:       broadcast ⟨VOTE_i, h, r, vote⟩   /* The execution of the broadcast has a cost c_send
          */
31:   Delivery step:
32:     delivery ⟨VOTE, h, r, v⟩   /* The process collects all the votes for the current height
          and round */
33:   Compute step:
34:     if |⟨VOTE, h, r, v⟩| ≥ v ∧ decidedValue = nil ∧ vote ≠ nil ∧ vote = v then
35:       decidedValue = v; exit
36:     else
37:       vote ← nil
38:       r ← r + 1
```

Note that, as defined in Section 2.3, the strategy σ_i^{send} depends on the knowledge the process has about the validity of the proposal. The strategy determines if the process chooses to send its vote for the proposal or not (line 24-30). If the processes choose to send a message for the proposal, it will pay a cost c_{send}.

Let us note that the rational player that did not check the validity of the block could decide an invalid value if more than v other processes have done the same and the proposed block is invalid. We also note that in our model, the Agreement property always holds, since at the end of each round, all rational processes have the same set of messages delivered.

We now define the game that represents the protocol.

2.3 Byzantine-Rational Game

Recall that out of the n players, $f \geq 1$ are Byzantine, while $n - f$ are rational. Each player i locally observes its own type, θ_i, which can be Byzantine ($\theta_i = \theta^b$) or rational ($\theta_i = \theta^r$).[1]

[1] If player's type was observable (i.e., if Byzantine players were detectable in advance) there would be a trivial solution to preclude them from harming the system: forbidding their participation.

Action space. As proposer, the player decides whether to propose a valid block or an invalid one. Then, at each round t, each player first decides whether to check the block's validity or not (at cost c_{check}), and second decides whether to send a message (at cost c_{send}) or not.

Information sets. At the beginning of each round $t > 1$, the information set of the player, h_i^t, includes the observation of the round number t, the player's own type θ_i, as well as the observation of what happened in previous rounds, namely (i) when the player decided to check validity, the knowledge of whether the block was valid or not, (ii) how many messages were sent, and (iii) whether a block was accepted or not. At round 1, h_i^1 only includes the player's private information about its own type, θ_i.

Then, in each round $t > 1$, the player decides whether to check the validity of the current block. At this point, denoting by b_t the block proposed at round t, when the player does not decide to check validity $\texttt{isValid}(b_t)$ is the null information set, while if the player decides to check, $\texttt{isValid}(b_t)$ is equal to 1 if the block is valid and 0 otherwise. So, at this stage the player information set becomes $H_i^t = h_i^t \cup \texttt{isValid}(b_t)$, which is h_i^t augmented with the validity information player i has about b_t, the proposed block.

Strategies. At each round $t \geq 1$, the strategy of player i is a mapping from its information set into its actions. If the agent is selected to propose the block, its choice is given by $\sigma_i^{\text{propose}}(h_i^t)$. Then, at the point at which the agent can decide to check block validity, its strategy is given by $\sigma_i^{\text{check}}(h_i^t)$. Finally, after making that decision, the player must decide whether to send a message or not, and that decision is given by $\sigma_i^{\text{send}}(H_i^t)$.

Reward and cost from adding blocks. In this paper we study the case in which, when a block is accepted, only the players which sent a message are rewarded (and receive R), as it is done in some blockchain systems (e.g [7]). In addition, we assume that when an invalid block is accepted, all rational players incur cost κ.

The reward R, given to the players when a block is accepted, is larger than the cost c_{check} of checking validity, which in turn is larger than the cost c_{send} of sending a message. Additionally, we assume that the reward obtained when a block is accepted is smaller than the cost κ of accepting an invalid block. That is, $\kappa > R > c_{check} > c_{send}$.

Objective of rational players. Let T be the endogenous round at which the game stops. If a block is accepted at round $t \leq n$, then $T = t$. Otherwise, if no block is accepted, $T = n$. In the latter case, the *termination* property is not satisfied.

At the beginning of the first round, the expected gain of rational player i is:

$$U_i = E\left[\begin{array}{c}\left(R * \mathbb{1}_{(\sigma_i^{\text{send}}(H_i^T)=1)} * \mathbb{1}_{(\text{block accepted at } T)}\right. \\ \left. -\kappa \mathbb{1}_{(\text{invalid block accepted})}\right) \\ -\sum_{t=1}^{T}\left(c_{check}\mathbb{1}_{\sigma_i^{\text{check}}(h_i^t)=1)} + c_{send}\mathbb{1}_{(\sigma_i^{\text{send}}(H_i^t)=1)}\right)\end{array} \Big| h_i^1\right],$$

where $\mathbb{1}_{(.)}$ denotes the indicator function, taking the value 1 if its argument is true, and 0 if it is false.

Then, at the beginning of round $t > 1$, if $T \geq t$, the continuation payoff of the rational player with information set h_i^t is

$$W_{i,t}(h_i^t) = E\left[\begin{array}{c}\left(R * \mathbb{1}_{(\sigma_i^{\text{send}}(H_i^T)=1)} * \mathbb{1}_{(\text{block accepted at } T)}\right. \\ \left. -\kappa \mathbb{1}_{(\text{invalid block accepted})}\right) \\ -\sum_{s=t}^{T}\left(c_{check}\mathbb{1}_{(\sigma_i^{\text{check}}(h_i^s)=1)} + c_{send}\mathbb{1}_{(\sigma_i^{\text{send}}(H_i^s)=1)}\right)\end{array} \Big| h_i^t\right],$$

Objective of Byzantine players. In the current paper, we assume the following: *Byzantine players 1) as proposers, propose invalid blocks, and 2) when receiving a proposed block, check the blocks' validity and send a message if and only if the block is invalid.*

We conjecture the above strategies will turn out to be the optimal strategies of the Byzantine players, minimizing $W_{i,t}$ in equilibrium.

Equilibrium concept. Since we consider a dynamic game, with asymmetric information, the relevant equilibrium concept is Perfect Bayesian Equilibrium [19], intuitively defined as follows:

A Perfect Bayesian equilibrium is such that all players 1) choose actions maximizing their objective function, 2) rationally anticipate the strategies of the others, and 3) draw rational inferences from what they observe, using their expectations about the strategies of the others and Bayes law, whenever it applies.

A Perfect Bayesian Equilibrium (PBE) is a Nash equilibrium [30], so players best-respond to one another. It imposes additional restrictions, to take into account the fact that the game is dynamic and that players can have private information, and therefore must draw rational inferences, from their observation of actions and outcomes. Rationality of inferences in PBE implies that (i) each player has rational expectations about the strategies of the others, and (ii) each player's beliefs are consistent with Bayes law, when computing probabilities conditional on events that have strictly positive probability on the equilibrium path. Perfection in PBE implies that, at each node starting a subgame the players' strategies form a Nash equilibrium of that subgame. In this context, to show that a strategy is optimal it is sufficient to show that it dominates any one-shot deviation [11].

We explore the behaviors of rational players that may not validate the block – because checking validity has a cost – and conditions (on the threshold ν and proportion of Byzantine players) under which rational players reach an equilibrium where both Validity and Termination properties (Section 2.2) are guaranteed. To do so, in Section 3, we study the equilibria that arise under different conditions.

3 EQUILIBRIA FOR RATIONAL PLAYERS

3.1 Equilibrium when $f \geq \nu$

When the number of Byzantine players is larger than ν, i.e., $f \geq \nu$, the validity property is not satisfied, since, when the first proposer is Byzantine, it proposes an invalid block, and that block is accepted, as all Byzantine players send messages in its favor. Against that backdrop, we characterize the strategies of the rational players and state the equilibrium outcome when $f \geq \nu$.

Proposition 3.1. *If $n - f \geq \nu + 1$ and $f \geq \nu$, there exists a Perfect Bayesian equilibrium in which the strategy of a rational player at any round is the following:*

- *As proposer, a rational player proposes a valid block.*
- *When receiving a proposed block, the rational players do not check the block validity but send a message.*

The first condition ($n - f \geq \nu + 1$) implies that, when all rational players but one send a message, they meet the majority threshold ν, so the block is accepted. The second condition ($f \geq \nu$) implies that, when all Byzantine players send a message, the block is accepted. Under these conditions, each rational player understands it is not pivotal: If the block is invalid, Byzantine players will send messages, so that the block will be accepted irrespective of the rational player's own action. Moreover, if the block is valid, Byzantine players will not send messages, but all the other rational players will, so that the block will be accepted irrespective of the rational player's action.

Thus rational players understand that they are not pivotal, and that whatever they do, given the equilibrium behavior of the other rational agents and of the Byzantine players; all blocks will be accepted. Consequently, they have no interest in checking the validity of the block. The only relevant comparison for them is between their expected gain when they send a message

$$R - c_{send} - \frac{f}{n}\kappa$$

and their expected gain when they do not send a message $-\frac{f}{n}\kappa$. Since, by assumption, $R > c_{send}$, rational players find it optimal to send a message. Finally note that, in the equilibrium of Proposition 3.1, a block is decided at round 1, so the *termination* property is satisfied, but, when the proposer is Byzantine, an invalid block is accepted, so the *validity* property is not satisfied.

3.2 Equilibria when $f < \nu$

Proposition 3.2. *When $f < \nu$ and $n - f \geq \nu$, there exists a Nash equilibrium in which rational players never check blocks' validity nor send messages, so that no block is ever accepted.*

Condition $f < \nu$ in Proposition 3.2 implies that Byzantine players cannot reach the threshold on their own. This precludes accepting invalid blocks. Therefore, the *validity* property is satisfied. Unfortunately, the condition also implies there exists an equilibrium in which the *termination* property also fails to hold. The intuition is the following:

In Proposition 3.2, each rational player anticipates that no other player will send a message when the block is valid.[2] In this context, each rational player knows that, if it were to send a message in favor of a valid block, it would be the only one to do so. Because the threshold ν is strictly larger than 1, the block would not be accepted. Therefore sending a message is a dominated action for the rational player. The equilibrium in Proposition 3.2 reflects that rational players' actions are strategic complements and they must coordinate on sending messages in order to have valid blocks accepted. Proposition 3.2 shows that, in equilibrium, there can be a coordination failure, such that no block is ever accepted.[3]

[2] Byzantine players send messages but only when the block is invalid.

[3] If $f = 0$, then, with $\nu = 1$, there exists a unique equilibrium, in which all players check validity and send a message iff the block is valid. In that equilibrium Validity and Termination are satisfied. But this obtains only if there are no Byzantine players. As soon as $f \geq 1$, if $\nu = 1$, Proposition 3.1 applies and validity is not satisfied.

Proposition 3.3. *When $f < \nu$ and $\nu < n - f - 1$, there exists a Perfect Bayesian equilibrium in which the strategy of a rational player at any round is the following:*

- *As proposer, a rational player proposes a valid block.*
- *When receiving a proposed block, the rational players do not check the block validity but send a message.*

As in Proposition 3.1, the rational players understand that they are not pivotal, and that whatever they do, given the equilibrium behavior of the other rational agents and of the Byzantine processes; all blocks will be accepted. Consequently, they have no interest in checking the validity of the block. The only relevant comparison for them is between their expected gain when they send a message $R - c_{send} - \frac{f}{n}\kappa$ and their expected gain when they do not send a message $-\frac{f}{n}\kappa$. Since, by assumption, $R > c_{send}$, rational players find it optimal to send a message. Finally note that, in the equilibrium of Proposition 3.3, a block is decided at round 1, so the *termination* property is satisfied, but, when the proposer is Byzantine, an invalid block is accepted, so the *validity* property is not satisfied. Note that the conditions of either Proposition 3.2 or of Proposition 3.3 imply that $f < \frac{n}{2}$, i.e, there is a strict majority of rational players. Yet, the propositions show that such majority is not enough to ensure both termination and validity.

While there exists equilibria in which either Termination (Proposition 3.2) is not verified or Validity (Proposition 3.3) is not verified, this does not necessarily imply there is no equilibrium that satisfies both properties. To have termination and validity, it must be that, in equilibrium, sufficiently many rational players find it in their own interest to check the validity of the block and to send messages in support of valid blocks. The problem is that some players might be tempted to free-ride, and let the others bear the cost of checking. To avoid this situation, it must be that (at least some) rational players anticipate they are pivotal, i.e., if they fail to check block validity and send messages in support of valid blocks, this may derail the player at their own expense.

To make this point, we look for an equilibrium in which some rational players check the validity of the block and send a message if and only if the block is valid, and this results in valid blocks being immediately accepted and invalid blocks being rejected. Before proving that such an equilibrium exists, we characterize the expected continuation payoff to which it would give rise.

Lemma 3.4. *Consider a candidate equilibrium in which some rational players check the validity of the block and send a message if and only if the block is valid, while the other rational players send messages without checking validity, and this results in valid blocks being immediately accepted and invalid blocks being rejected. In such an equilibrium, if it exists, the expected continuation payoff, at round t, of the rational players who are to check block validity is*

$$\pi_{check}(t) = R - c_{send} - \phi(t)c_{check},$$

while the expected continuation payoff, at round t, of the rational players who are not to check block validity is

$$\pi_{send}(t) = R - \psi(t)c_{send},$$

where $\phi(f) = 1$, $\psi(f+1) = 1$ and both ϕ and ψ satisfy property P defined below.

Definition 3.5. A function g satisfies property P, if $g(t) = 1 + \frac{f-t+1}{n-t+1} g(t+1), \forall t < f$.

In the candidate equilibrium, participants will reach a point at which the block is valid and all rational players send a message so that the block is accepted. This gives rise to a payoff $R - c_{send}$, the first part of $\pi_{check}(t)$. The second part of $\pi_{check}(t)$, $\phi(t)c_{check}$, is the expected cost of checking the block validity, where $\phi(t)$ is the expected number of times the player expects to check validity before a block is accepted. Similarly, in $\pi_{send}(t)$, $\psi(t)c_{check}$, is the expected cost of sending messages, where $\psi(t)$ is the expected number of times the player expects to send messages before a block is accepted.

Relying on Lemma 3.4, we now establish that our candidate equilibrium is indeed an equilibrium. To do so denote the highest index[4] of all Byzantine players by i_B.

Proposition 3.6. *When $f < \nu$ and $n - f > \nu$, if the cost κ of accepting an invalid block is large enough, in the sense that*

$$\kappa > \alpha(t)c_{check} - \beta(t)c_{send}, \forall t < f,$$

where

$$\alpha(t) = \frac{(n-t+1)\phi(t) - (f-t+1)\Pr(i_B \geq n-\nu+f+2 | T \geq t)\phi(t+1)}{(f-t+1)\Pr(i_B < n-\nu+f+2 | T \geq t)}$$

and

$$\beta(t) = \frac{\Pr(i_B \geq n-\nu+f+2 | T \geq t)}{\Pr(i_B < n-\nu+f+2 | T \geq t)},$$

and if the reward is large enough relative to the costs in the sense that

$$R \geq \max\left[\frac{n}{n-f}c_{send}, c_{send} + \frac{n}{n-f}c_{check}\right],$$

there exists a Perfect Bayesian equilibrium in which the strategy of rational players is the following:

- *As proposer, a rational player proposes a valid block.*
- *At any round $t \leq f$, when receiving a proposed block, (i) the rational players with index $i \in \{t, .. n-\nu+f+1\}$ check the block validity and send a message only if the block is valid, while (ii) the rational players with index $i \in \{n-\nu+f+2, ...n\}$ do not check the validity of the block but send a message.*
- *If round $t = f+1$ is reached, rational players send a message without checking if the block is valid. At this point the block is valid and accepted.*

Hence, in equilibrium, termination occurs no later than at round $f+1$.

On the equilibrium path, invalid blocks (proposed by Byzantine players) are rejected, while valid blocks (proposed by rational players) are accepted. This implies that, if round $t = f+1$ is reached, the players know that during all the previous (f) rounds the proposers were Byzantine (to draw this inference, the rational players use their anticipation that all participants play equilibrium strategies; hence the Perfect Bayesian nature of the equilibrium). Consequently, at round $f+1$, the proposer must be rational, and all players anticipate the proposed block is valid. So, no rational player needs to check the validity of the block but all send a message, which brings them expected gain equal to $R - c_{send}$. This is larger than their gain from deviating (e.g., by not sending a message or by checking the block.)

At previous rounds $t \leq f$, players know that all $t-1$ previous proposers were Byzantine and that there remains $f-t+1$ Byzantine players with index strictly larger than $t-1$ (as above, this rational inference is a feature of the Perfect Bayesian equilibrium we characterize). Do the equilibrium strategies of the rational players preclude acceptance of an invalid block by Byzantine processes? To examine that, consider the maximum possible number of messages that can be sent if the proposer is Byzantine. In equilibrium the $\nu - f - 1$ players with indexes strictly larger than $n - \nu + f + 1$ are to send a message without checking it. The worst case scenario (maximizing the number of messages sent when the block is invalid) is that none of these players are Byzantine. In that case, in equilibrium, the number of messages sent when the block is invalid is $f + (\nu - f - 1) = \nu - 1$, so that we narrowly escape validation of the invalid block. In contrast, if one of the rational players deviated from equilibrium and sent a message without checking the block, in the worst-case scenario, this would lead to accepting an invalid block. Thus, in that sense, the rational players with index strictly lower than $n - \nu + f + 1$ are pivotal. Hence, they check block validity, because, under the condition stated in the proposition, the cost of accepting an invalid block is so large that rational players do not want to run that risk.

Proof For clarity, we decompose the proof in 5 steps.

(1) The first step is to note that rational proposers strictly prefer to propose a valid block than an invalid one. This is because, when they follow their equilibrium strategy of proposing a valid block, it is accepted and the proposer gets $R - c_{check} - c_{send}$, while if they propose an invalid block, it is rejected, and we move to the next round, in which, in equilibrium, the player gets at most $R - c_{check} - c_{send}$ (and possibly less). Indeed, this player incurs the cost of checking validity at the next round, because the rational players who are not expected to check validity have indexes above $n - \nu + f + 1$, which are above $f + 1$, so that they do not get to propose blocks.

(2) The next step concerns the actions of the rational players when round $t = f + 1$ is reached. At that round, all players know the proposer must be rational and the proposed block valid. In equilibrium no rational checks validity but all send a message. Any other action would be dominated.

(3) The third step concerns the most relevant deviation, in which a rational player expected to check block validity fails to do so. If at round t a rational player supposed to check, deviates and sends a message without checking block validity, its expected continuation payoff is

$$\left(1 - \frac{f-(t-1)}{n-t+1}\right)(R - c_{send}) + \frac{f-(t-1)}{n-t+1}\Pr(i_B < n-\nu+f+1)(R - c_{send} - \kappa) + \frac{f-(t-1)}{n-t+1}\Pr(i_B \geq n-\nu+f+1)(\pi(t+1) - c_{send}).$$

The first term is the payoff obtained by the deviating rational player if the current block is valid, and therefore

[4] A player knows its index in the committee. Note that this is a common assumption in PBFT based blockchains.

immediately accepted. The second term is the payoff obtained by the deviating player when he was pivotal and triggered acceptance of an invalid block. To see this, consider the number of messages when the block is invalid, the rational player is deviating and the indexes of all the Byzantine players are strictly lower than $n-\nu+f+2$: f messages are sent by the Byzantine players, 1 message is sent by the deviating rational agent, $\nu-f-1$ messages are sent by the rational players with index above than or equal to $n-\nu+f+2$. The resulting total number of messages is ν and the block is accepted. The last term corresponds to the case in which the deviating rational player is not pivotal, and the invalid block is not accepted, so that we move to the next round.
Thanks to Lemma 3.4, we can substitute $\pi_{check}(t+1)$ by its value: $R-c_{send}-\phi(t+1)c_{check}$. After simplification, the expected continuation value of the deviating player is then:

$$(R-c_{send})-\frac{f-(t-1)}{n-t+1}\Pr(i_B<n-\nu+f+2|T\geq t)\kappa$$
$$-\frac{f-(t-1)}{n-t+1}\Pr(i_B\geq n-\nu+f+2|T\geq t)\left(\phi(t+1)c_{check}+c_{send}\right).$$

The equilibrium condition is that this deviation payoff must be lower than the equilibrium continuation payoff of the player: $R-c_{send}-\phi(t)c_{check}$. That is

$$\frac{f-(t-1)}{n-t+1}\Pr(i_B<n-\nu+f+2|T\geq t)\kappa>\phi(t)c_{check}$$
$$-\frac{f-(t-1)}{n-t+1}\Pr(i_B\geq n-\nu+f+2|T\geq t)\left(\phi(t+1)c_{check}+c_{send}\right).$$

Note that

$$\phi(t)\geq\frac{f-(t-1)}{n-t+1}\Pr(i_B\geq n-\nu+f+2|T\geq t)\phi(t+1),$$

by the definition of $\phi(t)$ this inequality is equivalent to

$$1+\frac{f-(t-1)}{n-t+1}\phi(t+1)\geq\frac{f-(t-1)}{n-t+1}\Pr(i_B\geq n-\nu+f+2|T\geq t)\phi(t+1),$$

which holds. Thus, we can write the equilibrium condition as

$$\kappa>\alpha(t)c_{check}-\beta(t)c_{send},\forall t<f,$$

as stated in the proposition.

(4) Other possible deviations for rational player supposed to check block's validity are easier to rule out:
First, the player could do nothing (neither check nor send). Relative to the equilibrium payoff, this deviation economizes the cost of checking (c_{check}). If the current proposer is Byzantine, the player then obtains the same payoff after a one shot deviation as on the equilibrium path ($\pi_{check}(t+1)$). If the current proposer is rational, the block gets accepted, but the player does not earn any reward. So the deviation is dominated if

$$\frac{n-f}{n-t+1}(R-c_{send})\geq c_{check},$$

which holds under the condition, stated in the proposition, that $R\geq\max\left[\frac{n}{n-f}c_{send},c_{send}+\frac{n}{n-f}c_{check}\right]$.
Second, the player could check the block validity, and then send a message irrespective of whether the block is valid or not. This would generate a lower payoff than the main deviation, shown above (in 3.) to be dominated.
Third, the player could check validity but then send no message. When the current proposer is Byzantine, this one-shot deviation yields the same payoff as the equilibrium strategy. When the current proposer is rational, this deviation yields a payoff of $-c_{check}$, which is lower than the equilibrium payoff $R-c_{send}-c_{check}$.
Fourth, the player could check the block's validity and send a message only if the block is invalid, which is dominated.

(5) Finally turn to deviations of rational players supposed to send messages without checking blocks' validity.
First, consider the possibility to abstain from sending a message. This economizes the costs c_{send}, but, in case the block is valid and accepted, this implies the agent loses the reward R. Therefore, the deviation is dominated if

$$\frac{n-f}{n-t+1}R\geq c_{send},$$

which holds under the condition, stated in the proposition, that $R\geq\max\left[\frac{n}{n-f}c_{send},c_{send}+\frac{n}{n-f}c_{check}\right]$.
Second, consider the possibility of checking validity and sending a message only for valid blocks. This deviation would imply the agent would have to incur the cost of checking (c_{check}), but it would economize the cost of sending a message when the block is invalid. So the deviation is dominated if

$$c_{check}\geq\frac{f-t+1}{n-t+1}c_{send},$$

which holds, since by assumption $c_{check}\geq c_{send}$.
Other deviations, such as checking validity but never sending messages, or checking validity and always sending messages, or checking validity and sending only if the block is invalid, are trivially dominated.

□*Proposition* 3.6

4 CONCLUSION AND FUTURE WORK

In this paper, we study the resilience of multi-agent Byzantine consensus-based blockchains by modeling them as a coordination game between rational and Byzantine players. We derived the conditions (on the majority threshold and the proportion of Byzantine players) under which the Validity and Termination properties of the consensus are guaranteed in equilibrium or not, assuming that we always have agreement. In future work, we will extend the analysis to more general Byzantine strategies and rational agents' preferences, costs and rewards. Note that our work is the first game theoretical analyses that considers the combination of rational and Byzantine players in consensus-based blockchains. Previous game theoretical works in these settings considered either all players are rational, or the combination of correct (follow the protocol specification) and rational.

REFERENCES

[1] Ittai Abraham, Lorenzo Alvisi, and Joseph Y. Halpern. 2011. Distributed computing meets game theory: combining insights from two fields. *SIGACT News* 42, 2 (2011), 69–76.

[2] Ittai Abraham, Danny Dolev, Rica Gonen, and Joseph Y. 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, PODC 2006, Denver, CO, USA, July 23-26, 2006*. 53–62.

[3] Ittai Abraham, Danny Dolev, and Joseph Y. Halpern. 2019. Distributed Protocols for Leader Election: A Game-Theoretic Perspective. *DISC 2013 and ACM Trans. Economics and Comput.* 7, 1 (2019), 4:1–4:26.

[4] Ittai Abraham, Dahlia Malkhi, Kartik Nayak, Ling Ren, and Alexander Spiegelman. 2016. Solidus: An Incentive-compatible Cryptocurrency Based on Permissionless Byzantine Consensus. *CoRR* abs/1612.02916v1 (2016).

[5] Yehuda Afek, Yehonatan Ginzberg, Shir Landau Feibish, and Moshe Sulamy. 2014. Distributed computing building blocks for rational agents. In *ACM Symposium on Principles of Distributed Computing, PODC '14, Paris, France, July 15-18, 2014*. 406–415.

[6] Amitanand S. Aiyer, Lorenzo Alvisi, Allen Clement, Michael Dahlin, Jean-Philippe Martin, and Carl Porth. 2005. BAR fault tolerance for cooperative services. In *Proceedings of the 20th ACM Symposium on Operating Systems Principles 2005, SOSP 2005, Brighton, UK, October 23-26, 2005*. 45–58.

[7] Yackolley Amoussou-Guenou, Antonella Del Pozzo, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. 2018. Correctness of Tendermint-Core Blockchains. In *22nd International Conference on Principles of Distributed Systems, OPODIS 2018, December 17-19, 2018, Hong Kong, China*. 16:1–16:16.

[8] Emmanuelle Anceaume, Antonella Del Pozzo, Romaric Ludinard, Maria Potop-Butucaru, and Sara Tucci-Piergiovanni. 2019. Blockchain Abstract Data Type. In *The 31st ACM on Symposium on Parallelism in Algorithms and Architectures, SPAA 2019, Phoenix, AZ, USA, June 22-24, 2019*. 349–358.

[9] Christian Badertscher, Juan A. Garay, Ueli Maurer, Daniel Tschudi, and Vassilis Zikas. 2018. But Why Does It Work? A Rational Protocol Design Treatment of Bitcoin. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*. 34–65.

[10] Bruno Biais, Christophe Bisière, Matthieu Bouvard, and Catherine Casamatta. 2019. The blockchain folk theorem. *The Review of Financial Studies* (2019).

[11] David Blackwell. 1965. Discounted Dynamic Programming. *The Annals of Mathematical Statistics* 36, 1 (1965), 226–235. https://doi.org/10.1214/aoms/1177700285

[12] Christian Cachin, Klaus Kursawe, Frank Petzold, and Victor Shoup. 2001. Secure and efficient asynchronous broadcast protocols (extended abstract. In *in Advances in Cryptology: CRYPTO 2001*. Springer, 524–541.

[13] Miguel Castro and Barbara Liskov. 1999. Practical Byzantine Fault Tolerance. In *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA, February 22-25, 1999*. 173–186.

[14] T. Crain, V. Gramoli, M. Larrea, and M. Raynal. 2017. (Leader/Randomization/Signature)-free Byzantine Consensus for Consortium Blockchains. http://csrg.redbellyblockchain.io/doc/ConsensusRedBellyBlockchain.pdf. (2017).

[15] Bernardo David, Peter Gazi, Aggelos Kiayias, and Alexander Russell. 2018. Ouroboros Praos: An Adaptively-Secure, Semi-synchronous Proof-of-Stake Blockchain. In *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*. 66–98.

[16] C. Decker, J. Seidel, and R. Wattenhofer. 2016. Bitcoin Meets Strong Consistency. In *Proceedings of the 17th International Conference on Distributed Computing and Networking Conference (ICDCN)*.

[17] I. Eyal, A. E. Gencer, E. Gün Sirer, and R. van Renesse. 2016. Bitcoin-NG: A Scalable Blockchain Protocol. In *13th USENIX Symposium on Networked Systems Design and Implementation, (NSDI)*.

[18] Ittay Eyal and Emin Gün Sirer. 2018. Majority is not enough: bitcoin mining is vulnerable. *Commun. ACM* 61, 7 (2018), 95–102.

[19] Drew Fudenberg and Jean Tirole. 1991. Perfect Bayesian equilibrium and sequential equilibrium. *Journal of Economic Theory* 53, 2 (1991), 236 – 260. https://doi.org/10.1016/0022-0531(91)90155-W

[20] Juan A. Garay, Jonathan Katz, Ueli Maurer, Björn Tackmann, and Vassilis Zikas. 2013. Rational Protocol Design: Cryptography against Incentive-Driven Adversaries. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*. 648–657.

[21] 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, October 28-31, 2017*. 51–68.

[22] Adam Groce, Jonathan Katz, Aishwarya Thiruvengadam, and Vassilis Zikas. 2012. Byzantine Agreement with a Rational Adversary. In *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*. 561–572.

[23] Joseph Y. Halpern and Xavier Vilaça. 2016. Rational Consensus: Extended Abstract. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing, PODC 2016, Chicago, IL, USA, July 25-28, 2016*. 137–146.

[24] E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford. 2016. Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing. In *Proceedings of the 25th USENIX Security Symposium*.

[25] L. Lamport, R. Shostak, and M. Pease. 1982. The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems* 4, 3 (July 1982), 382–401.

[26] Anna Lysyanskaya and Nikos Triandopoulos. 2006. Rationality and Adversarial Behavior in Multi-party Computation. In *Advances in Cryptology - CRYPTO 2006, 26th Annual International Cryptology Conference, Santa Barbara, California, USA, August 20-24, 2006, Proceedings*. 180–197.

[27] Mohammad Hossein Manshaei, Murtuza Jadliwala, Anindya Maiti, and Mahdi Fooladgar. 2018. A Game-Theoretic Analysis of Shard-Based Permissionless Blockchains. *IEEE Access* 6 (2018), 78100–78112.

[28] 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, October 24-28, 2016*. 31–42.

[29] S. Nakamoto. 2008. Bitcoin: A Peer-to-Peer Electronic Cash System. https://bitcoin.org/bitcoin.pdf. (2008).

[30] John Nash. 1951. Non-Cooperative Games. *Annals of Mathematics* 54, 2 (1951), 286–295. http://www.jstor.org/stable/1969529

[31] Fahad Saleh. 2019. *Blockchain Without Waste: Proof-of-Stake*. SSRN Scholarly Paper ID 3183935. Social Science Research Network, Rochester, NY.

[32] Ayelet Sapirshtein, Yonatan Sompolinsky, and Aviv Zohar. 2016. Optimal Selfish Mining Strategies in Bitcoin. In *Financial Cryptography and Data Security - 20th International Conference, FC 2016, Christ Church, Barbados, February 22-26, 2016, Revised Selected Papers*. 515–532.

[33] 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, PODC 2019, Toronto, ON, Canada, July 29 - August 2, 2019*. 347–356.