TRAP: The Bait of Rational Players to Solve Byzantine Consensus

Alejandro Ranchal-Pedrosa
University of Sydney
Sydney, Australia
alejandro.ranchalpedrosa@sydney.edu.au

Vincent Gramoli
University of Sydney and EPFL
Sydney, Australia
vincent.gramoli@sydney.edu.au

ABSTRACT

It is impossible to solve the Byzantine consensus problem in an open network of n participants if only $2n/3$ or less of them are correct. As blockchains need to solve consensus, one might think that blockchains need more than $2n/3$ correct participants. But it is yet unknown whether consensus can be solved when less than $2n/3$ participants are correct and k participants are rational players, which misbehave if they can gain the loot. Trading correct participants for rational players may not seem helpful to solve consensus since rational players can misbehave whereas correct participants, by definition, cannot.

In this paper, we show that consensus is actually solvable in this model, even with less than $2n/3$ correct participants. The key idea is a *baiting strategy* that lets rational players pretend to misbehave in joining a coalition but rewards them to betray this coalition before the loot gets stolen. We propose TRAP, a protocol that builds upon recent advances in the theory of accountability to solve consensus as soon as $n > \max\left(\frac{3}{2}k+3t, 2(k+t)\right)$: by assuming that private keys cannot be forged, this protocol is an equilibrium where no coalition of k rational players can coordinate to increase their expected utility regardless of the arbitrary behavior of up to t Byzantine players.

Finally, we show that a baiting strategy is necessary and sufficient to solve this, so-called *rational agreement* problem. First, we show that it is impossible to solve this rational agreement problem without implementing a baiting strategy. Second, the existence of TRAP demonstrates the sufficiency of the baiting strategy. Our TRAP protocol finds applications in blockchains to prevent players from disagreeing, that could otherwise lead to "double spending".

CCS CONCEPTS

• **Theory of computation → Algorithmic game theory**; • **Security and privacy → Distributed systems security**; • **Computing methodologies → Distributed algorithms**.

KEYWORDS

Blockchain, consensus, game theory, robustness, fault tolerance

ACM Reference Format:
Alejandro Ranchal-Pedrosa and Vincent Gramoli. 2022. TRAP: The Bait of Rational Players to Solve Byzantine Consensus. In *Proceedings of the 2022 ACM Asia Conference on Computer and Communications Security (ASIA CCS '22), May 30-June 3, 2022, Nagasaki, Japan.* ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3488932.3517386

ASIACCS '22, May 30–June 3, 2022, Nagasaki, Japan

ACM ISBN 978-1-4503-9140-5/22/05…$15.00
https://doi.org/10.1145/3488932.3517386

1 INTRODUCTION AND BACKGROUND

Consider n players, each with some initial value. Solving the Byzantine consensus problem consists of designing a protocol guaranteeing that the $n-t$ non-Byzantine players agree by outputting the same value and despite the presence of up to t arbitrarily behaving Byzantine players. Standard cryptography, which permits oblivious signed transfers and assumes computationally bounded players, has recently been used to build undeniable proofs identifying the players that led the system to a disagreement [12, 13]. Although this construction has not been used in the game theoretical context, one can intuitively see its application to implement a *baiting strategy* that incentivizes rational players to solve blockchain consensus: the idea is to reward rational players to pretend to join a coalition in order to deceit the coalition by exposing undeniable *proofs-of-fraud* (PoFs).

Placed in the game theory context, one can see a consensus protocol among rational players as a Nash equilibrium, however, a Nash equilibrium only prevents one rational player from increasing its utility by deviating solely, but it fails at preventing multiple rational players from increasing their utility by colluding and by all deviating together. The resilience against this type of coalition, mentioned originally in the 50s [8], is needed to solve consensus despite a coalition of k players. Ben-Porath [10] shows that one can simulate a Nash equilibrium with a central trusted mediator provided that there is a punishment strategy to threaten rational players in case they deviate and Heller [29] strengthens Ben-Porath's result to allow coalitions. Abraham et al. [2] applied these results to secret sharing in the fully distributed setting, by showing that one can simulate a mediator with cheap talks and assuming the same standard cryptography we assume. A (k,t)-punishment strategy guarantees that if up to k rational players deviate, then more than t non-deviating players, by playing the punishment strategy, can lower the utility of these rational players.

Another challenge when making consensus Byzantine fault tolerant is for the equilibrium to be immune to t Byzantine players that act arbitrarily or whose utility functions are unknown. Abraham et al. [2] were the first to formalize k-resilience, t-immunity and ϵ-(k,t)-robustness. A protocol is a k-resilient equilibrium if no rational coalition of size up to k can increase their utility by deviating in a coordinated way. A protocol is t-immune if the expected utility of the $n-t$ non-faulty players is not decreased regardless of the arbitrary behavior of up to t Byzantine players. A protocol is ϵ-(k,t)-robust if no coalition of k rational players can coordinate to increase their expected utility by ϵ regardless of the arbitrary behavior of up to t Byzantine players, even if the Byzantine players join their coalition, where ϵ accounts for the (small) probability of breaking the cryptography.

Considering rational players alongside the well-known Byzantine faults to ensure the agreement property of the consensus problem allows us to evaluate protocols against coalitions of players that are incentivized to deviate for their own benefit or to break the system, rather than just because of their faulty nature. This *rational agreement* problem finds application in blockchains where participants can be modeled as players incentivized to gain valuable cryptocurrency assets. In 2020, these assets incentivized players to deviate from their blockchain protocol by forcing a disagreement to fork the blockchain, leading to a double spending of US$70,000 in Bitcoin Gold[1] and US$5.6 million in Ethereum Classic[2]. Solving rational agreement is key to avoid these losses because it ensures that players agree on a unique block at each index of the chain, thus preventing these forks from being exploited to double spend. Interestingly, some blockchains already require participants to deposit cryptocurrency assets in the form of staking and these deposits could be used by the consensus protocol to incentivize their owners to behave. Unfortunately, these blockchains fork in the presence of rational players and we are not aware of any solution to this rational agreement problem.

1.1 Our result

In this paper, we show that a *baiting strategy*, that incentivize rational players to bait deviating players into a trap, is necessary and sufficient to solve this *rational agreement* problem by offering a consensus protocol that is robust to a coalition of up to k rational players and t Byzantine players. Our first contribution is thus to formalize the notion of a baiting strategy and show that this baiting strategy is necessary to solve the rational agreement problem. Motivated by open networks, where blockchains typically operate, and where the synchrony assumption [19] in unrealistic, we then solve the rational agreement problem without assuming a known bound on the delay of every message. As solving the consensus problem is impossible with complete asynchrony [21], we consider the partially synchronous model where there is an unknown bound on the delay of messages [19]. This solution bypasses the well-known requirement of $2n/3$ correct participants [32], by solving consensus when $n > \max(\frac{3}{2}k + 3t, 2(k + t))$. For example if $n = 7$ players, then our solution solves consensus with only 4 correct participants, hence tolerating $k = 1$ rational player and $t = 2$ Byzantine players.

We implement this solution in a new protocol, called Trap (**T**ackling **R**ational **A**greement through **P**ersuasion), that is ϵ-(k, t)-robust when $n > \max(\frac{3}{2}k + 3t, 2(k + t))$. Trap rewards a single player to expose its coalition by generating proofs-of-fraud with the Polygraph accountable consensus protocol [13]. Making initially colluding players decide on whether to betray the coalition regardless of the coalition behavior is analogous to reducing the extensive-form game into a normal-form game for this particular decision. In addition, if the reward for exposing the coalition is greater than the individual payoff for causing a disagreement, rational players find that the coalition strategy is strictly dominated by the baiting strategy in the extensive-form game. This shows similarities with the prisoner's dilemma, in which all rational players prefer to betray the coalition than to collude.

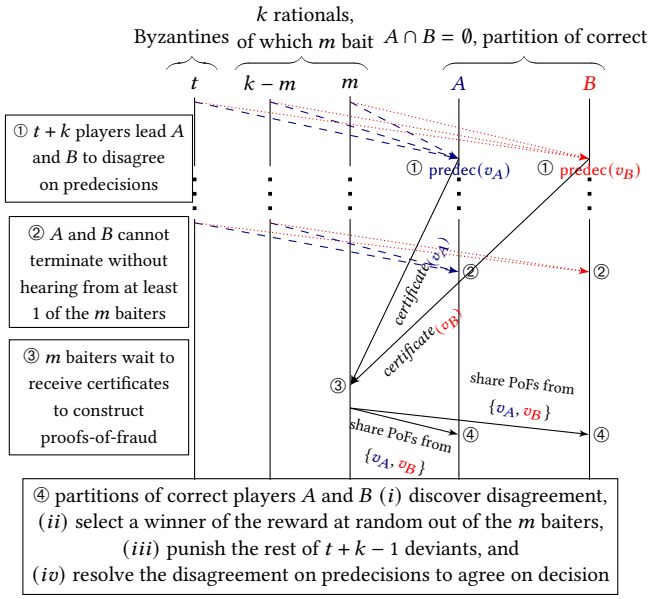

Figure 1: Example execution of the Trap protocol. First, ① all t Byzantine and k rational players collude to cause a disagreement on the output of the accountable consensus protocol, resulting in A and B predeciding different outputs. Then, ② m of the k rational players commit to bait while executing the BFTCR protocol, preventing A and B from deciding their disagreeing predecisions. As such, ③ the m baiters wait until they receive proof of the disagreement on predecisions, to then ④ prove the disagreement committing to and revealing the proofs-of-fraud in the BFTCR protocol. Hence, neither A nor B decide their conflicting predecisions, but instead reward one of the m baiters, punish the rest of $t + k - 1$ players responsible for the disagreement on predecisions, and resolve the disagreement, deciding one of v_A or v_B, or, depending on the application, merging both.

More specifically, our protocol "pre-decides" the decisions from Polygraph [13] that it extends with the *Byzantine Fault Tolerant Commit-Reveal protocol (BFTCR)*, which consists of two reliable broadcasts and one additional broadcast. As we show in Figure 1, by offering a reward the Trap protocol can convince m rational players to betray the coalition after they helped cause a disagreement on predecisions. First, the coalition exposes itself causing a disagreement on predecisions. Second, the m rational players from the coalition that decide to betray are enough to pause termination of the BFTCR protocol. Third, these players can wait to gather enough PoFs of the disagreement on predecisions, which they will get by the property of accountability (since they caused a disagreement on the output of the Polygraph protocol). Fourth and finally, once they get PoFs to prove the disagreement to correct players and get the reward, these m players commit and reveal the PoFs by

[1] https://news.bitcoin.com/bitcoin-gold-51-attacked-network-loses-70000-in-double-spends/
[2] https://news.bitcoin.com/5-6-million-stolen-as-etc-team-finally-acknowledge-the-51-attack-on-network/

sharing them during the BFTCR protocol, after which one of them will be selected at random to get the reward. We detail further this example in Appendix A.

Adding this BFTCR phase ensures the existence of a baiting strategy (baiting-dominance) and that the protocol still solves agreement even after playing the baiting strategy (baiting-agreement). We also add an additional property, lossfree-reward, which states that the increase in utility for baiting rational players comes at no cost to non-deviating players. For this purpose, we introduce a deposit per player, so that the system can always pay the reward by taking the deposits of the proven coalition at no cost for non-deviating players.

1.2 Related work

Considering fault tolerant distributed protocols as games requires to cope with a mixture of up to k rational players and t faulty players. The idea of mixing rational players with faulty players has already been extensively explored in the context of secret sharing and multi-party computation [2, 16, 22, 33]. In particular, the central third-party mediator that is typically relied upon was implemented with synchronous *cheap talks* [2], that are communications of negligible cost through private pairwise channels. This extension was indeed illustrated with an ϵ-(k, t)-robust secret sharing protocol where $n > k + 2t$. It was later shown [1] that mediators could be implemented with asynchronous cheap talks in an ϵ-(k, t)-robust protocol when $n > 3(k + t)$. This adaptation makes it impossible to devise even a 1-immune protocol that would solve the consensus problem [32] as the communication model becomes asynchronous [21]. In this paper, we focus instead on the partially synchronous model, where the bound on the delay of messages is unknown [19], to design a protocol that solves consensus among n players, where up to t are Byzantine players and k are rational players.

Consensus has been explored in the context of game theory. Some works focused on the conditions under which termination and validity is obtained for a non-negligible cost of communication and/or local computation [7, 38], without considering the incentives for rational players to cause a disagreement. This incentive is quite apparent in the blockchain context, where Bitcoin users hacked the network to double spend by simply leading sets of players to a disagreement (or fork) for long enough[3]. Some results consider the problem of consensus in the presence of rational players but do not consider failures [24]. Leader election [3], which can be used to solve consensus indirectly, and consensus proposals [26] focus on ensuring fairness defined as all players having an equal probability of their proposal being decided. Some proposals study consensus and mix faulty players with rational players [5, 9], however, they consider the synchronous communication model.

Several research results focus more particularly on agreement, with some deriving from the BAR (Byzantine-Altruistic-Rational) model. However, these works considered either no Byzantine players [20, 25], no coalitions of rational players [6], synchrony [25, 27, 28, 39] or solution preference [27]. By assuming a larger payoff for agreeing than for disagreeing, solution preference requires that rational players never have an incentive to sabotage agreement. To the best of our knowledge, we present the first work that considers bounds for the robustness of agreement against coalitions of Byzantine and rational players in partial synchrony.

The baiting strategy that we introduced to reward traitors of a coalition is very similar to the betrayal used in the context of verifiable cloud-computing for counter-collusion contracts, assuming that the third party hosting the contracts is trusted [18]. Our BFTCR protocol presents a novel implementation to select the winner of the baiting reward without a trusted third party.

There are a number of advantages of BFTCR compared to other state-of-the-art protocols. One may think that a solution similar to submarine commitments [11] would work as well by, for example, hiding the proofs-of-fraud in a decision. However, such a solution does not prevent Byzantine players from always hiding in a submarine commitment their proofs-of-fraud, and revealing them only if a rational player reveals their submarine commitment, which can act as a deterrent for rational players to not betray the coalition. Additionally, other protocols based on zero-knowledge proofs [31] explicitly reveal the existence of an information to prove, which gives an additional advantage to other players in the coalition to also claim the same knowledge.

State-of-the-art verifiable secret-sharing (VSS) schemes [4, 17, 30, 37] are not a good fit either, since there are no secret-sharing schemes in partial synchrony that tolerate coalitions of size greater than a third of the participants. To the best of our knowledge, BFTCR is the first protocol that implements baiting strategies for consensus tolerating coalitions of up to k rational and t Byzantine players as long as $n > \max\left(\frac{3}{2}k + 3t, 2(k + t)\right)$.

1.3 Roadmap

The rest of the paper is structured as follows: Section 2 presents our model and preliminary definitions, Section 3 introduces the definition of a baiting strategy and shows that it is impossible to solve the rational agreement problem without a baiting strategy, in Section 4 we present the Trap protocol and its correctness, and we finally conclude in Section 5.

2 PRELIMINARIES

We consider a partially synchronous communication network, in which messages can be delayed by up to a bound that is unknown. For this purpose, we adapt the synchronous and asynchronous models of Abraham et al. [1, 2] to partial synchrony. We consider a game played by a set N of $|N| = n$ players, each of type in $\mathcal{T} = \{\text{Byzantine, rational, correct}\}$. The game is in *extensive form*, described by a game tree whose leaves are labeled by the utilities u_i of each player i. We introduce the scheduler as an additional player that will model the delay on messages derived from partial synchrony. We assume that players alternate making moves with the *scheduler*: first the scheduler moves, then a player moves, then the scheduler moves and so on. The scheduler's move consists of choosing a player i to move next and a set of messages in transit to i that will be delivered just before i moves (so that i's move can depend on all the messages i delivers). Every non-leaf node is associated with either a player or the scheduler. The scheduler is bound to two constraints. First, the scheduler can choose to delay any message *msg* up to a bound, known only to the scheduler,

[3] https://www.cnet.com/news/hacker-swipes-83000-from-bitcoin-mining-pools/.

before which he must have chosen all recipients of *msg* to move and provided them with this message, so that they deliver it before making a move. Second, the scheduler must eventually choose all players that are still playing. That is, if player i is playing at time e, then i is chosen to play at time $e' \geq e$.

Each player i has some *local state* at each node, which translates into the initial information known by i, the messages i sent and received at the time that i moves, and the moves that i has made. The nodes where a player i moves are further partitioned into *information sets*, which are sets of nodes in the game tree that contain the same local state for the same player i, in that i cannot distinguish them. We assume that the scheduler has complete information, so that the scheduler's information sets consist of the singletons.

Since we do not assume synchrony, we need our game to be able to continue even if a faulty player decides not to reply. As such, w.l.o.g. we assume that players that decide not to play will at least play the *default-move*, which consists of notifying the scheduler that this player will not move, so that the game continues with the scheduler choosing the next player to move. Thus, in every node where the scheduler is to play a move, the scheduler can play any move that combines a player and a subset of messages that such player can deliver before playing. Then, the selected player moves, after which the scheduler selects again the next player for the next node, and the messages it receives, and so on. The scheduler alternates thus with one player at each node down a path in the game tree until reaching a leaf. A *run* of the game is then a downward path in the tree from the root to a leaf.

Strategies. We denote the set of actions of a player i (or the scheduler) as A_i (or A_s), and a strategy σ_i for that set of actions is denoted as a function from i's information sets to a distribution over the actions. We denote the set of all possible strategies of player i as $\mathcal{S}_i$. Let $\mathcal{S}_I = \Pi_{i \in I} \mathcal{S}_i$ and $A_I = \Pi_{i \in I} A_i$ for a subset $I \subseteq N$. Let $\mathcal{S} = \mathcal{S}_N$ with $A_{-I} = \Pi_{i \notin I} A_i$ and $\mathcal{S}_{-I} = \Pi_{i \notin I} \mathcal{S}_i$. A *joint strategy* $\vec{\sigma} = (\sigma_0, \sigma_1, ..., \sigma_{n-1})$ draws thus a distribution over paths in the game tree (given the scheduler's strategy σ_s), where $u_i(\vec{\sigma}, \sigma_s)$ is player's i expected utility if $\vec{\sigma}$ is played along with a strategy for the scheduler σ_s. A strategy θ_i *strictly dominates* τ_i for i if for all $\vec{\phi}_{-i} \in \mathcal{S}_{-i}$ and all strategies σ_s of the scheduler we have $u_i(\theta_i, \vec{\phi}_{-i}, \sigma_s) > u_i(\tau_i, \vec{\phi}_{-i}, \sigma_s)$.

Given some desired functionality $\mathcal{F}$, a *protocol* is the recommended joint strategy $\vec{\sigma}$ whose outcome satisfies $\mathcal{F}$ for all strategies σ_s of the scheduler, and an *associated game* Γ for that protocol is defined as all possible deviations from the protocol [2]. In this case, we say that the protocol $\vec{\sigma}$ *implements* the functionality. Note that both the scheduler and the players can use probabilistic strategies.

Failure model. We set $t_0 = \lceil \frac{n}{3} \rceil - 1$ for the rest of this paper and k players out of n can be rational while $t \leq t_0$ can be Byzantine; the rest of the players are correct. *Correct players* follow the protocol: the expected utility of correct player i is greater than 0 for any run in which the outcome satisfies consensus, and 0 for any other run. *Rational players* can deviate to follow the strategy that yields them the greatest expected utility at any time they are to move, while *Byzantine players* can deviate in any way, even not replying at all (apart from notifying the scheduler that they will not move). Rational players have greater utility for outcomes in which they caused a disagreement than from outcomes that satisfy consensus, but have no interest in deviating from consensus for anything else, in that they prefer to terminate and to guarantee validity. We will detail further the utilities of rational players in Section 4.3.

We assume that if a coalition manages to cause a disagreement, then it obtains a payoff of at most $\mathcal{G}$, which we call the *total gain*. Nevertheless, this total gain may be, for example, the entire market value of the system. In a payment system application in which players agree on a set of transactions to be decided, the total gain $\mathcal{G}$ is exactly the sum of all the amounts spent in all transactions. We also assume, w.l.o.g., that a coalition with k rational players and t Byzantine players split equally the total gain into k parts, which we call the *gain* $g = \mathcal{G}/k$, that is, Byzantine players are willing to give all the total gain from causing a disagreement to the rational players that collude (to incentivize the deviation for these rational players). Note that a protocol that tolerates a maximum gain $\mathcal{G}$ equally split into k parts also tolerates any gain such that the maximum share of the split is $\mathcal{G}/k$, but we assume the equal split for ease of exposition. We speak of the *disagreeing* strategy as the strategy in which players collude to produce a disagreement, and of a coalition *disagreeing* to refer to a coalition that plays the disagreeing strategy. A disagreement of consensus can mean two or more disjoint groups of non-deviating players deciding two or more separate, conflicting decisions [36]. For ease of exposition, we consider in this work only disagreements into two values. Nonetheless, if the size of the coalition is less than half the total number of players $k + t < n/2$ (as is the case for the work that we present) then the coalition can only cause a disagreement into two values [36], whereas greater sizes of a coalition can cause disagreements into multiple values [34].

We let rational players in a coalition and Byzantine players (in or outside the coalition) know the types of all players, so that they know which players are the other Byzantine players, rational players and correct players, while the rest of the players only know the upper bounds on the number of rational and Byzantine players, i.e., k and t respectively, and their own individual type (that is, whether they are rational, Byzantine or correct).

Cheap talks. As we are in a fully distributed system, without a trusted central entity like a mediator, we assume *cheap-talks*, that is, private pairwise communication channels. We also assume negligible communication cost through these channels. Non-Byzantine players are also only interested in reaching consensus, and not in the number of messages exchanged. Similarly, we assume the cost of performing local computations (such as validating proposals, or verifying signatures) to be negligible.

Cryptography. We require the use of standard cryptography, for which we reuse the assumptions of Goldreich et al. [23]: polynomially bounded players and the enhanced trapdoor permutations. In practice, these two assumptions mean that players can sign unforgeable messages, and that they can perform oblivious transfer. Each player has a public key and a private key, and public keys are common knowledge.

Robustness. Given that a Nash equilibrium only protects against single-player deviations, and our distributed system may be susceptible of a coalition of k rational and t Byzantine players, it is important to consider tolerating multi-player deviations. We thus restate Abraham's et al. [2] definitions of t-immunity, ϵ-(k, t)-robustness

and the most recent definition of k-resilient equilibrium [1]. The notion of k-resilience is motivated in distributed computing by the need to tolerate a coalition of k rational players that can all coordinate actions. A joint strategy is k-resilient if not all rational members of a coalition of size at most k can gain greater utility by deviating in a coordinated way.

Definition 2.1 (k-resilient equilibrium)**.** A joint strategy $\vec{\sigma} \in \mathcal{S}$ is a *k-resilient equilibrium* (resp. *strongly k-resilient equilibrium*) if, for all $K \subseteq N$ with $|K| \leq k$, all $\vec{\tau}_K \in \mathcal{S}_K$, all strategies σ_s of the scheduler, and for some (resp. all) $i \in K$ we have $u_i(\vec{\sigma}_K, \vec{\sigma}_{-K}, \sigma_s) \geq u_i(\vec{\tau}_K, \vec{\sigma}_{-K}, \sigma_s)$.

The notion of t-immunity is motivated in distributed algorithms by the need to tolerate t Byzantine players. An equilibrium $\vec{\sigma}$ is t-immune if non-Byzantine players still prefer to follow $\vec{\sigma}$ despite the deviations of up to t Byzantine players.

Definition 2.2 (t-immunity)**.** A joint strategy $\vec{\sigma} \in \mathcal{S}$ is t-immune if, for all $T \subseteq N$ with $|T| \leq t$, all $\vec{\tau} \in \mathcal{S}_T$, all $i \notin T$ and all strategies of the scheduler σ_s, we have $u_i(\vec{\sigma}_{-T}, \vec{\tau}_T, \sigma_s) \geq u_i(\vec{\sigma}, \sigma_s)$.

A joint strategy is an ϵ-(k,t)-robust equilibrium if no coalition of k rational players can coordinate to increase their expected utility by ϵ regardless of the arbitrary behavior of up to t Byzantine players, even if the Byzantine players join their coalition. We illustrate it however with ϵ because of the use of cryptography, that is, in order to account for the (negligible) probability of the coalition breaking cryptography, as was done previously [2]:

Definition 2.3 (ϵ-(k,t)-robust equilibrium)**.** A joint strategy $\vec{\sigma} \in \mathcal{S}$ is an ϵ-(k,t)-robust (resp. strongly ϵ-(k,t)-robust) equilibrium if for all $K, T \subseteq N$ such that $K \cap T = \emptyset, |K| \leq k$, and $|T| \leq t$, for all $\vec{\tau}_T \in \mathcal{S}_T$, for all $\vec{\phi}_K \in \mathcal{S}_K$, for some (resp. all) $i \in K$, and all strategies of the scheduler σ_s, we have $u_i(\vec{\sigma}_{-T}, \vec{\tau}_T, \sigma_s) \geq u_i(\vec{\sigma}_{N-(K \cup T)}, \vec{\phi}_K, \vec{\tau}_T, \sigma_s) - \epsilon$. We speak instead of a (k,t)-robust equilibrium if $\epsilon = 0$.

We use a recent definition of k-resilient equilibrium [1], which slightly differs from the definition of an ϵ-(k,t)-robust equilibrium. We define here strong resilience and strong robustness to refer to the stronger versions of these properties [2]. Byzantine fault tolerance in distributed computing is equivalent to our definition of t-immunity in game theory.

Given some game Γ and desired functionality $\mathcal{F}$, we say that a protocol $\vec{\sigma}$ is a k-resilient protocol for $\mathcal{F}$ if $\vec{\sigma}$ implements $\mathcal{F}$ and is a k-resilient equilibrium. For example, if $\vec{\sigma}$ is a k-resilient protocol for the consensus problem, then in all runs of $\vec{\sigma}$, every non-deviating player terminates and agrees on the same valid value. We extend this notation to t-immunity and ϵ-(k,t)-robustness. The required functionality of this paper is thus reaching agreement.

Punishment strategy. We also restate the definition of a punishment strategy [2] as a threat that correct and rational players can play in order to prevent other rational players from deviating. The punishment strategy guarantees that if k rational players deviate, then $t+1$ players can lower the utility of these rational players by playing the punishment strategy.

Definition 2.4 ((k,t)-punishment strategy)**.** A joint strategy $\vec{\rho}$ is a (k,t)-punishment strategy with respect to $\vec{\sigma}$ if for all $K, T, P \subseteq N$ such that K, T, P are disjoint, $|K| \leq k, |T| \leq t, |P| > t$, for all $\vec{\tau} \in \mathcal{S}_T$, for all $\vec{\phi}_K \in \mathcal{S}_K$, for all $i \in K$, and all strategies of the scheduler σ_s, we have $u_i(\vec{\sigma}_{-T}, \vec{\tau}_T, \sigma_s) > u_i(\vec{\sigma}_{N-(K \cup T \cup P)}, \vec{\phi}_K, \vec{\tau}_T, \vec{\rho}_P, \sigma_s)$.

Intuitively, a punishment strategy represents a threat to prevent rational players from deviating, in that if they deviate, then players in P can play the punishment strategy $\vec{\rho}$ and the deviating rational players decrease their utility with respect to following $\vec{\sigma}$. For example, crime sentences are an effective punishment strategy against committing crimes. Not terminating a protocol if just one player deviates can also be a punishment strategy against deviating from the protocol.

Accountability. Previous work introduced signatures in consensus protocol messages, guaranteeing that for a disagreement to occur, at least $t_0 + 1$ players must sign conflicting messages, and once these messages are discovered by a correct player, such player can prove the fraudsters to the rest of correct players through *Proofs-of-Fraud (PoFs)* [12, 34]. We also adapt to this model the property of accountability, recently defined for consensus [12, 13]:

Definition 2.5 (accountability)**.** Let $\vec{\sigma}$ be a protocol that implements agreement. Suppose that a disagreement takes place, then $\vec{\sigma}$ is accountable if all correct players will eventually gather enough proof that at least $t_0 + 1$ players deviated to cause the disagreement.

Rational agreement. In the remainder, we are interested in proposing a consensus protocol that is immune to up to t_0 Byzantine failures and robust to a coalition of up to k rational and t Byzantine players, so we restate the Byzantine consensus problem [32] in the presence of rational players: The *Byzantine consensus problem* is, given n players, each with an initial value, to ensure (i) *agreement* in that no two non-deviating players decide different values; (ii) *validity* in that the decided value has to be proposed; and (iii) *termination* in that eventually every non-deviating player decides.

Definition 2.6 (Rational Agreement)**.** Consider a system with n players, a protocol $\vec{\sigma}$ solves the rational agreement problem if it implements consensus, and is t_0-immune and ϵ-(k,t)-robust for some $k, t > 0$ such that $n \leq 3(k+t)$.

3 RATIONAL AGREEMENT IMPOSSIBILITY WITHOUT A BAITING STRATEGY

In this section, we introduce a baiting strategy as a particular case of punishment strategy and show that it is necessary to devise a consensus protocol robust to a coalition of k rational players and t Byzantine players.

Our solution to agreement in the presence of rational and Byzantine players, presented in Section 4.3, consists of rewarding rational players for betraying the coalition. One may wonder whether rewarding rational players in a coalition is the only way to obtain ϵ-(k,t)-robustness that tolerates coalitions of size $n \leq 3(k+t)$ in partial synchrony. To demonstrate the need for a reward, we first formalize a type of (k,t)-punishment strategy, which we call a (k,t,m)-baiting strategy. A (k,t,m)-baiting strategy is a $(k-m,t)$-punishment strategy such that $k \geq m > 0$, and these m rational players prefer to actually play the baiting strategy than to deviate with the rest of the players in the coalition. That is, m players of the coalition have to play the baiting strategy for it to succeed, and

at least m rational players in the coalition prefer to play the baiting strategy than to deviate with the coalition. An example is offering a crime reduction for a criminal to cooperate with law enforcement into catching the criminal group to which it belongs.

Definition 3.1 ((k, t, m)-baiting strategy)**.** A joint strategy $\vec{\eta}$ is a (k, t, m)-baiting strategy with respect to a strategy $\vec{\sigma}$ if $\vec{\eta}$ is a $(k - m, t)$-punishment strategy with respect to $\vec{\sigma}$, with $0 < m \leq k$ and for all $K, T, P \subseteq N$ such that $K \cap T = \emptyset$, $|P \cap K| \geq m$, $P \cap T = \emptyset$, $|K \backslash P| \leq k - m, |T| \leq t, |P| > t$, for all $\vec{\tau} \in \mathcal{S}_T$, all $\vec{\phi}_{K \backslash P} \in \mathcal{S}_{K \backslash P} - \{\vec{\sigma}_K\}$, all $\vec{\theta}_P \in \mathcal{S}_P$, all $i \in P$, and all strategies of the scheduler σ_s, we have:

$$u_i(\vec{\sigma}_{N-(K \cup T \cup P)}, \vec{\phi}_{K \backslash P}, \vec{\tau}_T, \vec{\eta}_P, \sigma_s) \geq u_i(\vec{\sigma}_{N-(K \cup T \cup P)}, \vec{\phi}_{K \backslash P}, \vec{\tau}_T, \vec{\theta}_P, \sigma_s).$$

Additionally, we speak of a strong (k, t, m)-*baiting strategy* in the particular case where for all rational coalitions $K \subseteq N$ such that $|K| \leq k$, $|K \cap P| \geq m$ and all $\vec{\phi}_{K \backslash P} \in \mathcal{S}_{K \backslash P}$ we have: $\sum_{i \in K} u_i(\vec{\sigma}_{N-(K \cup P)}, \vec{\phi}_{K \backslash P}, \vec{\eta}_P, \sigma_s) \leq \sum_{i \in K} u_i(\vec{\sigma}, \sigma_s)$.

A baiting strategy illustrates a situation where at least m rational players in the coalition may be interested in baiting other $k + t - m$ rational and Byzantine players into a trap: the $k + t$ of them collude to deviate initially, just so that these m players can prove such deviation by playing the baiting strategy, and get a reward for exposing this deviation. Such a strategy has a significant impact in a protocol to implement agreement. A strong baiting strategy defines a baiting strategy in which the fact that m deviating players play the baiting strategy does not yield greater payoff to the entire coalition as a whole (if such coalition was made only by rationals), compared to following the protocol. This prevents a coalition of rational players from colluding together so as to play the baiting strategy on themselves only with the purpose of splitting the baiting reward among the colluding members. Notwithstanding, neither a baiting strategy nor a strong baiting strategy show that if these m players play the baiting strategy, then the protocol implements the desired functionality. We illustrate the efficacy of baiting strategies to influence the outcome of a protocol in the example of the rational generals, shown in Figure 2.

Impossibility result. The reason why a (k, t, m)-baiting strategy is relevant to the consensus problem is that without such a strategy it is not possible to obtain a consensus protocol that is (k, t)-robust where $k > 0$. We show this result in Theorem 3.2. The proof is similar to that of the impossibility of t-immune consensus under partial synchrony for $t > t_0$ [19], since a partition of rational and Byzantine players can exploit two disjoint partitions of correct players to lead them to different decisions. Let us recall that we do not assume solution preference, and thus the payoffs from a disagreement can be singificantly greater than those of agreeing for rational players. For the proof of Theorem 3.2, we first show the more general proof of Lemma 3.1.

Lemma 3.1. *It is impossible to obtain a protocol $\vec{\sigma}$ that implements agreement, is t_0-immune and (k, t)-robust, $k \geq 0$ and $t = \max(t_0 - k + 1, 0)$ unless there is a (k, t, m)-baiting strategy with respect to $\vec{\sigma}$, for $m > \frac{k+t-n}{2} + t_0$.*

Proof. We refer to Dwork et al.'s [19] work for the impossibility of increasing $t > t_0$ and obtaining agreement (i.e., for $k = 0$). For $k > 0$ with $t \leq t_0$, assume the contrary: let $\vec{\sigma}$ be a protocol such that there is no (k, t, m)-baiting strategy with respect to $\vec{\sigma}$ and $\vec{\sigma}$ is (k, t)-robust, for $t = max(t_0 - k + 1, 0)$, $k > 0$. Since the protocol is t_0-immune and it works under partial synchrony, the protocol must not require more than $n - t_0$ players participating in

Rational generals example. We illustrate the intuition behind baiting strategies with an example inspired from the Byzantine generals problem [32], that we refer to as the 'rational generals' problem: suppose $n = 7$ Ottoman generals need to agree on whether to attack or retreat. If all generals agree on attacking, they will succeed, if they agree on retreat, they can succeed another day. However, if only some of the generals attack, they will lose. There are two Byzantine generals, i.e., $t = 2$, whose goal is for the Ottomans to disagree on their decision for them to lose, and another rational general, i.e., $k = 1$, who has been offered a bribe $\mathcal{G}$ in order to contribute to the disagreement, but who is willing to betray the Byzantines for a greater income from the Ottomans. Because of accountability, the generals will eventually be able to track the disagreement to both the t Byzantine and k rational generals, but by then the k rational generals will be enjoying its reward $\mathcal{G}$ in Constantinople, out of reach.

The generals suspect that there might be a bribed rational general ($k = 1$). In an attempt from them to make the rational general talk, they offer a reward $\mathcal{R} > \mathcal{G}$ as a bounty for proving the fraud of every other Byzantine and rational general, that is, if the rational general reveals its identity and that of the t Byzantine with proofs, then this rational general is spared and rewarded with $\mathcal{R}$, while the t Byzantine generals lose all of their capital (i.e., properties and savings) that they own in the Ottoman empire. In this case, the rational general sees a greater incentive to expose both himself and the Byzantine generals. This is an example of a baiting strategy. Additionally, the Ottoman generals will pay $\mathcal{R}$ with the capital taken from the t Byzantine generals, so the Ottoman empire will not even pay for the reward.

Notice that Ottoman generals must guarantee to the rational general that they will recognize him as the first to expose the coalition (and the only rightful owner of the reward), so that the rational general is not influenced by a threat from the Byzantine generals to steal the reward if he betrays the coalition. That is, the rational general will only bait the coalition if the protocol ensures that the Byzantine generals will not be able to steal the reward from the rational general after seeing that he betrayed the coalition. This is in order to prevent the Byzantine generals from rushing to bait as soon as they learn the rational general is starting to bait, creating a situation in which both Byzantine and rational generals seem to be legitimate baiters of the coalition.

In the extensive game, this means that the rational general must first behave and make moves as if he would cause the disagreement. Then, the rational general will only bait if he gets both enough evidence of the fraud of the deviants and assurance that the Byzantine generals will not outpace him and steal the reward.

Figure 2: Rational generals example.

it in order to take a decision, or else the Byzantine players could prevent termination. Consider a partition of the network between 4 disjoint subsets $N = K \cup A \cup B \cup F$, where K are the rational players (there is at least one), F are the Byzantine players, i.e., $|F| + |K| = t + k \geq t_0 + 1$, and A and B are the rest of the players such that $|A| + |B| \leq n - t_0 - 1$ and both $|A| + |F| + |K| \geq n - t_0$ and $|B| + |F| + |K| \geq n - t_0$ hold (recall $t_0 = \lceil \frac{n}{3} \rceil - 1$). Let $\vec{\theta}$ be the strategy in which the rational players in K deviate with Byzantine players in F and achieve a disagreement between players in A and players in B. If the players in F and K are all Byzantine and rational players, then such a disagreement is always possible and the utility for each rational player is, by definition of the model, greater than that of reaching agreement. Notice also that since $t = \max(t_0 - k + 1, 0)$, if $m > \frac{k+t-n}{2} + t_0$ rational players do not deviate to cause such disagreement, we have that at least one of $|A| + |F| + |K| - m < n - t_0$ and $|B| + |F| + |K| - m < n - t_0$ holds, or both: for this value of m the deviants cannot cause a disagreement. However, this is not true if instead $m \leq \frac{k+t-n}{2} + t_0$. It follows that it is necessary to encourage at least $m > \frac{k+t-n}{2} + t_0$ rational players to not deviate into causing a disagreement, which means, by definition, that a (k, t, m)-baiting strategy is necessary. □

Theorem 3.2. *It is impossible to obtain a protocol $\vec{\sigma}$ that implements rational agreement unless there is a (k, t, m)-baiting strategy with respect to $\vec{\sigma}$, for $m > \frac{k+t-n}{2} + t_0$.*

Proof. By definition, every (k, t)-robust protocol for $n \leq 3(k+t)$ must also be (k, t)-robust, for some $k \geq 0$ and $t = \max(t_0 - k + 1, 0)$. Therefore it derives from Lemma 3.1. □

Theorem 3.2 shows the need for a baiting strategy to solve rational agreement. In Section 4.2 we show the implementation of an additional phase to an accountable consensus protocol in order to provide the functionality of a baiting strategy. In Section 4.3 we illustrate the values of a reward and deposit per player to make a strong baiting strategy that at least m rational players will play.

4 TRAP: REACHING RATIONAL AGREEMENT

In this section, we present the Trap (**T**ackling **R**ational **A**greement through **P**ersuasion) protocol, the first protocol to solve the rational agreement problem. The Trap protocol comprises three components:

(1) A financial component, consisting of a deposit per player $\mathcal{L}$, taken at the start of the protocol from each participating player, and a reward $\mathcal{R}$, which is given to a player in the event that it provides PoFs for a disagreement on predecisions.
(2) An accountable consensus component, that pre-decides outputs from an accountable consensus protocol.
(3) A baiting component, embodied in a novel Byzantine Fault Tolerant *commit-reveal* (BFTCR) protocol that executes after the accountable consensus protocol. This component terminates either deciding one output (predecision) of the accountable consensus protocol, or resolving a disagreement on predecisions by rewarding one of the deviating players that exposed the disagreement and punishing the rest of deviating players.

We first provide an overview of the properties that we aim at for the Trap protocol in Section 4.1, and the possible runs of the game that derive from implementing a strong baiting strategy for the rational agreement problem with the aforementioned components. We then introduce and prove the correctness of the baiting component, the BFTCR protocol, in Section 4.2. Finally, we analyze the financial component, that is, the specific values of reward and deposits, in Section 4.3. The accountable consensus component can be any accountable consensus protocol [12–14, 35], and thus we treat this component as a black box, for the sake of generality.

4.1 Overview: consensus with a baiting strategy

We proved in Section 3 that we need a baiting strategy for a protocol to solve the rational agreement problem.

Before we present the implementation of such a baiting strategy in Section 4.2, with additional configurations of the required deposits and reward sizes in Section 4.3, we present in this section the basics of our baiting strategy. For this purpose, we focus first on the properties that we aim at for such a baiting strategy. Then, we showcase all the possible runs of a protocol for consensus that provides such a strong baiting strategy.

Given a protocol $\vec{\sigma}$ that implements accountable consensus and is t_0-immune, we will extend it to implement the rational agreement problem, in that we will prove the three following properties:

- *Baiting-dominance*: There is a (k, t, m)-baiting strategy $\vec{\eta}$ with respect to $\vec{\sigma}$, for $m > \frac{k+t-n}{2} + t_0$.
- *Baiting-agreement*: $\vec{\eta}$ implements agreement.
- *Lossfree-reward*: $\vec{\eta}$ is a strong baiting strategy.

Baiting-dominance states the necessary condition that a baiting strategy exists, while baiting-agreement guarantees that playing such a baiting strategy still leads to agreement. Lossfree-reward guarantees that such a baiting strategy is a strong baiting strategy. Coming back to the rational generals example of Figure 2, baiting-dominance states the existence of the reward for the rational general, baiting-agreement guarantees that generals will still decide whether to attack or retreat after paying the reward to the rational general, and lossfree-reward guarantees that only the slashed capital of the Byzantine generals will be used to pay the reward to the rational general.

Reward for baiting. Since the protocol is accountable, we add a *baiting reward* $\mathcal{R}$ for player i if i can prove to the rest of the players that a coalition of at least $t_0 + 1$ players are trying to cause a disagreement, but before they succeed at causing the disagreement. If multiple players are eligible for the baiting reward, then only one is chosen at random to win the reward, and the rest are treated as fraudsters that did not bait. We select the winner at random in an additional *winner consensus* in which the winner is decided from among the proposed candidates to win from correct replicas in this winner consensus. We explain further the winner consensus later in this section. Players can prove that a coalition is trying to cause a disagreement through PoFs which undeniably show two conflicting messages signed by the same set of players. The reward is only given to i if i exposes this coalition before the coalition causes the disagreement (i.e., before both partitions of correct players decide different decisions).

Funding the reward with deposits. we require all players to place a minimum *deposit* $\mathcal{L}$. We also require such deposit to be big enough so that the deposit taken from the exposed coalition is enough to pay the reward, satisfying lossfree-reward. Our goal is to set $\mathcal{R}$ and $\mathcal{L}$ so that we implement a baiting strategy for a set M of rational players in the coalition, such that if others in the coalition bait, then for all $i \in M$, player i is better off also trying to bait and getting the reward, while if the rest of the players in the coalition do not bait, then if i baits then i gets the greatest expected utility that it can in that information set. We analyze in Theorem 4.4 the required values for such deposit and reward necessary to incentivize at least $|M| = m > \frac{k+t-n}{2} + t_0$ rational players in a coalition to follow a baiting strategy, depending on the size $k + t$ of the coalition and on the maximum total gain from disagreeing $\mathcal{G}$. For now, however, let us ignore the values of $\mathcal{L}$ and $\mathcal{R}$ and focus on the protocol that solves the rational agreement problem, by assuming that these values of $\mathcal{L}$ and $\mathcal{R}$ are enough to make $m > \frac{k+t-n}{2} + t_0$ rational players bait the coalition, instead of terminating a disagreement. We will come back to specify proper values for $\mathcal{L}$ and $\mathcal{R}$ in Section 4.3. If these PoFs expose at least $t_0 + 1$ players including the winner of the baiting reward $\mathcal{R}$, then the t_0 (or more) remaining colluding players lose the deposit amount $\mathcal{L}$.

Dominating disagreements. We explore here the possible runs, assuming that we already have such a baiting strategy, and what each of these runs means for the payoffs of a rational player i:

(1) Rational players including i contribute to reaching agreement and follow the protocol $\vec{\sigma}$, getting some utility $u_i(\vec{\sigma}_{-T}, \vec{\tau}_T) \geq \epsilon$ where $\epsilon > 0$.

(2) Some rational players collude with i and deviate to disagree, playing strategy $\vec{\phi}$ with some Byzantine players T and other rational players K such that $|K \cup T| \geq n/3$, $K \cap T = \emptyset$, obtaining utility $u_i(\vec{\sigma}_{N-K-T}, \vec{\phi}_{K \cup T}) = g$.

(3) Player i deviates to bait other rational players into colluding with some Byzantine players such that $|K \cup T| \geq n/3, K \cap T = \emptyset$, and this deviation consists of playing strategy $\vec{\eta}$ to expose the colluding players via PoFs and obtain the baiting reward. As a result, player i obtains utility $u_i(\vec{\sigma}_{N-K-T}, \vec{\phi}_{K \cup T-M}, \vec{\eta}_M) = p(m)\mathcal{R} - q(m)\mathcal{L}$, where M is the set of players of the coalition that bait, i.e., $i \in M$, with $|M| = m$. $p(m) = 1/m$ represents the probability of winning the reward, while $q(m) = 1 - p(m) = (m-1)/m$ the probability of not winning it after baiting.

(4) Player i deviates to disagree only to suffer a trap baited by another rational (or group of rational players), obtaining utility $u_i(\vec{\sigma}_{N-K-T}, \vec{\phi}_{K \cup T-M}, \vec{\eta}_M) \leq -\mathcal{L}$.

(5) In any run where the protocol does not terminate, player i obtains negative utility.

(6) Player i contributes to reaching agreement but a coalition causes a disagreement. In this case, i is one of the victims of a disagreement (for example, a double-spending). Hence, i obtains negative utility.

Notice that the runs 4, 5 and 6 are strictly dominated by run 1 (following the protocol). Our goal is to make runs represented by 3 runs that also implement agreement and that strictly dominate runs represented by 2.

4.2 Baiting component: the BFTCR protocol

In this section, we present the first implementation of a baiting strategy for rational agreement. As such, we extend an accountable consensus protocol with a Byzantine Fault Tolerant *commit-reveal* (BFTCR) phase in order to solve consensus even if there is a disagreement at consensus level, if at least m rational players decide to betray the coalition in exchange for trying to win a reward. We show in Algorithm 1 the BFTCR phase. As such, we speak of a *predecision* for a decision of the accountable consensus protocol, whereas a *decision* now refers to the outcome of the BFTCR protocol. The BFTCR phase consists of 5 main parts:

(1) A reliable broadcast, in which players share their encrypted commitment (line 15),
(2) a second reliable broadcast, in which players share the first $(n - t_0)$ encrypted commitments that they delivered in the first reliable broadcast (line 19),
(3) a regular broadcast, in which players share the key to reveal their commitment (line 23),
(4) an additional consensus to select the winner of the reward, if some players reveal a list of PoFs (line 39), and
(5) a slashing of the deposits from the fraudsters, payment of the reward to the winner and resolution of the disagreement on predecisions (line 40).

Commit and reveal. The purpose of the first group of reliable broadcasts is to reliably broadcast the encrypted PoFs, should a player own them, or an encrypted hash of a predecision otherwise. We say that the *commitment* is the encrypted content that each player decides to broadcast in this first reliable broadcast. In line 19 each player i then starts the second reliable broadcast by broadcasting a list of the first $(n - t_0)$ delivered commitments that i delivered in the first reliable broadcast. The purpose of the calls to broadcast on lines 23 and 25 is to deliver the keys to decrypt the encrypted messages. A player i thus *reveals* his commitment by broadcasting the key. A player i decrypts the commitment of player j in line 26. Then, player i adds this decrypted message to the list of decided hashes in lines 28 to 31, or to the list of PoFs received in lines 33 to 35.

Termination. The BFTCR phase of the Trap protocol terminates in one of two ways:

- either there is no disagreement on predecisions, and then the protocol terminates when at least $(n - t_0)$ messages are decrypted with the same hash of the predecisions in line 31;
- or some players reveal a disagreement on predecisions through PoFs, and then the protocol terminates when at least $t_0 + 1$ messages are decrypted (without counting players that are proven to be false through a PoF) with a reward to a chosen baiter and a punishment to the remaining players that are listed in the PoFs from lines 36 to 40.

Note that accountability does not guarantee that a baiter will gather enough PoFs before a disagreement takes place. We prove that baiters will gather enough PoFs before a disagreement takes place as part of the proof of Theorem 4.2. The idea is that m rational players will wait to receive enough PoFs to be able to commit to bait, where m is big enough to prevent termination of either of the partitions of correct players.

Valid candidates of the winner consensus. We define a *valid candidate* to win the reward as a member of a deviating coalition that committed to bait the coalition (by sending a commitment to a list of PoFs of the coalition in line 15) independently of whether other m players of the coalition also committed to bait, for $m > \frac{k+t-n}{2} + t_0$. The objective of the BFTCR protocol is to distinguish valid candidates from players who try to win the reward only after they learn that the disagreement will not succeed. A correct player i considers a baiter j as a valid candidate if i can see j's commitment to bait in at least $t_0 + 1$ messages from the second reliable broadcast. We refer to this $t_0 + 1$ messages as a *proof-of-baiting* (PoB). The BFTCR protocol selects the winner of the bait among the list of valid candidates by executing an additional consensus, in the call to select_winner in line 39, in which all participating players propose the PoFs they know about and the valid candidates, along with the PoBs. We detail further this call later in this section.

Note that a rational player i that commits to bait a coalition may deviate from Algorithm 1 in order to hinder other deviants from becoming valid candidates after i reveals its commitment. This is because this way i maximizes its chances of winning the reward (by minimizing the number of valid candidates for the reward). This is an expected deviation of a baiting rational player, which consists on waiting to deliver as many messages from the second reliable broadcast as possible from both partitions of correct players that suffered the disagreement on predecisions, and we show the correctness of this approach as part of the proof of Theorem 4.4.

Correctness and randomness of the winner consensus. We show in Theorem 4.5 that no deviating player can win the reward without being a valid candidate, i.e., no player can bait and win the reward after learning that other m (or more) players baited. Additionally, note that the winner consensus solves consensus for $n > 9/5(k + t)$ because at least $t_0 + 1$ provably fraudulent players of the coalition will not participate in it, as has already been shown [34], and we consider $n > 2(k + t)$. That is, at most $n' = n - (t_0 + 1) < 2n/3$ players participate in the winner consensus. Since the maximum coalition size is $k + t < n/2$, then the remaining players of the coalition that could participate in the winner consensus are $t' = n/2 - (t_0 + 1) < n/6$, and thus $t' < n'/3$ and the winner consensus solves consensus. Furthermore, the winner consensus only terminates once at least $n - t'$ proposals have been decided, which can be optimized through a democratic consensus protocol [12, 13, 15, 34]. Finally, after $n - t'$ proposals are decided upon, the participants execute an iteration of a random beacon that tolerates $t' < n'/3$ Byzantine faults [17, 30, 37], in order to select the winner of the baiting reward randomly from among any of the valid candidates that were in any of the decided proposals.

Following the winner consensus, in line 40, fraudsters are punished and the baiter is rewarded, respectively. The call to resolve(...) resolves the two disagreeing predecisions by deterministically choosing one of them (i.e., lexicographical order) or, depending on the application, merging both.

Resolving a disagreement on consensus predecisions with BFTCR. It is clear that if there is no disagreement on the predecisions, the BFTCR phase will terminate and satisfy consensus. We consider here the output of the BFTCR phase in the case where there is a disagreement into two predecisions. We speak of a disagreement on predecisions being *finalized* if it becomes a disagreement on decisions (that is, on the output of the BFTCR phase). We will show in Theorem 4.2 that if $m > \frac{k+t-n}{2} + t_0$ rational players commit to bait instead of finalizing the disagreement on predecisions, then the Trap protocol still satisfies consensus. For this purpose, we define $m(k, t) = \lfloor \frac{k+t-n}{2} + t_0 \rfloor + 1$ (i.e., the smallest natural value that satisfies $m > \frac{k+t-n}{2} + t_0$). Then, we first show in Lemma 4.1 that if $m(k, t)$ rational players bait, then the only possible outcome is to resolve a disagreement on predecisions.

Algorithm 1 BFT commit-reveal protocol for (correct) player i

```
1:  State:
2:      enc_msgs, list of delivered encrypted messages from the first group reliable
3:          broadcasts, initially ∅
4:      list_enc_msgs, list of delivered encrypted messages by other players from the
5:          first group of reliable broadcasts, initially ∅
6:      decrypted_msgs, list of delivered decrypted messages from the first group of
7:          reliable broadcasts, initially ∅
8:      {RB_j^1}_{j=0}^n, the first group of reliable broadcasts where j is the source
9:      {RB_j^2}_{j=0}^n, the second group of reliable broadcasts where j is the source
10:     hashes, a dictionary where keys are hashes and values are integers, initially it
11:         does not contain any key or value
12:     local_hash, local hash of the predecided value, according to this player
13:     POF_received, boolean, initially False
14:     i, i_msg, i_key, i_enc_msg, player's id, message, key, and encrypted message

15: RB_i^1.start(i_enc_msg)                          ▷ start first group of reliable broadcasts

16: Upon RB-delivering enc_msg from reliable broadcast RB_j^1:
17:     enc_msgs[j] ← enc_msgs
18:     if (size(enc_msgs) ≥ n − t_0) then
19:         RB_i^2.start(enc_msgs)

20: Upon RB-delivering enc_msgs_j from reliable broadcast RB_j^2:
21:     list_enc_msgs[j] ← enc_msgs_j
22:     if (size(list_enc_msgs) ≥ n − t_0 and size(enc_msgs) ≥ n − t_0) then
23:         broadcast(i_key, i)          ▷ reveal i's commitment by broadcasting decryption key

24: Upon delivering key from j and RB-delivering from RB_j^1 and RB_j^2 :
25:     broadcast(key, j)
26:     decrypted_msgs[j] ← decrypt(enc_msgs, key)                          ▷ decrypt it
27:     if (decrypted_msgs[j].type = HASH) then      ▷ if it is the hash of a predecision
28:         hash ← decrypted_msgs[j].get_hash()
29:         hashes[hash] += 1                                               ▷ add to count
30:         if (hashes[hash] ≥ n − t_0 and local_hash = hashes[hash]) then
31:             decide(hash)          ▷ if count for this hash reaches threshold, then decide it
32:     else if (decrypted_msgs[j].type = POFS) then                        ▷ if instead list of PoFs
33:         PoFs ← decrypted_msgs[j].get_PoFs()
34:         if (verify(PoFs)) then list_PoFs[j] ← PoFs                      ▷ verify PoFs are valid
35:             POF_received ← True
36:     if (POF_received) then
37:         msgs_filtered ← keys(decrypted_msgs) \ keys(PoFs)
38:         if (size(msgs_filtered) ≥ t_0 + 1) then                         ▷ winner consensus
39:             baiter, frauds, predec_1 predec_2 ← select_winner(list_enc_msgs, lPoFs)
40:             punish(frauds); reward(baiter); resolve(predec_1, predec_2)
```

Lemma 4.1. *Let n players play the associated game of the* Trap *protocol $\vec{\sigma}$, out of which k can be rational and t Byzantine, with $n > 2(k+t)$. Suppose a run in which a coalition causes a disagreement on predecisions, and consider the start of the BFTCR phase. Then, if $m(k, t)$ rational players of the coalition commit to bait then the only possible outcome is to pay the reward and resolve the disagreement on predecisions.*

Proof. First, we show that $m(k, t)$ deviating players committing to bait suffices to prevent the disagreement on predecisions to be

finalized in a disagreement on decisions. This is analogous to the proof of Lemma 3.1. Then, we show that if $m(k,t)$ players commit to bait, then the BFTCR phase safely terminates resolving predecisions, with all correct players that start the winner consensus terminating it and agreeing. Finally, we show that deviating players cannot get the reward and also cause a disagreement, i.e., if one player terminates the winner consensus then all correct players start it.

Suppose two predecisions v_A, v_B that two partitions of players not in the coalition A and B predecided, such that $A \cap B = \emptyset$, and $|A|+|B|+k+t \leq n$. For A to decide v_A (resp. B to decide v_B), players in A (resp. B) must be able to decide without hearing from players in B (resp. A). Therefore, $|A|+k+t \geq n-t_0$ and also $|B|+k+t \geq n-t_0$ to finalize the disagreement. We consider now how many m rational players out of k must bait (i.e., must not contribute to finalizing the disagreement) for a disagreement to necessarily fail. This value must be such that $|A|+(k-m)+t < n-t_0$ and same for B's partition, which solves to $m > \frac{k+t-n}{2} + t_0$ (analogously to Lemma 3.1).

Then, we recall that the BFTCR phase resolves predecisions, rewards and punishes players if at least $t_0 + 1$ players have been exposed through PoFs. Thus, every non-deviating player can ignore messages received from a set containing at least $t_0 + 1$ players. All non-deviating players eventually converge to the same set of detected fraudsters [34], as all correct players broadcast the PoFs they hear from and update their detected fraudsters accordingly. As such, let F represent the set of detected fraudsters, then for all $|F| \in [t_0 + 1, k + t]$ it follows that $n'/3 > k + t - |F|$ for $n' = n - |F|$, and thus the winner consensus tolerates deviations from the rest of rational and Byzantine players not yet detected.

Finally, we show that if the reward is paid, then it is not possible to cause a disagreement at decision level. We have shown in the previous paragraph that all non-deviating players that execute the winner consensus terminate agreeing. We must thus prove that if a correct player terminates the winner consensus, then no correct player can terminate deciding a predecision without executing the winner consensus. Since $n'/3 > k + t - |F|$, the winner consensus terminates with the participation of just $2n'/3$ players, of which at least $2n'/3 - (k+t-|F|)$ are correct. Since there are $n-k-t$ correct players in total, if the winner consensus terminates for some correct player, then there are at most $c = n-k-t-(2n'/3-(k+t-|F|)) = (n-|F|)/3$ correct players that have neither learned about the disagreement nor executed the winner consensus yet. Thus, for these remaining correct players to not be able to decide without executing the winner consensus, it is necessary that $c + t + k < n - t_0 \iff t + k < n/2$.

Hence, as long as at least $m(k,t) = \lfloor \frac{k+t-n}{2} + t_0 \rfloor + 1$ rational players play the baiting strategy, the only possible outcome is for one of them to get the reward, and to resolve the disagreement on predecisions.

□

Theorem 4.2 (baiting-agreement)**.** *Let n players play the associated game of the* Trap *protocol $\vec{\sigma}$, of which k can be rational and t Byzantine, with $n > 2(k+t)$. Suppose that $m(k,t) = \lfloor \frac{k+t-n}{2} + t_0 \rfloor + 1$ rational players in the coalition play the baiting strategy committing to bait if they participate in a disagreement on predecisions. Then the* Trap *protocol solves rational agreement.*

Proof. The proof of t_0-immunity follows from the proof of Polygraph's t_0-immunity and the fact that the additional BFTCR phase consists of two Byzantine fault tolerant reliable broadcasts and one additional broadcast per player, terminating each of them if $n - t_0$ players follow the protocol.

For ϵ-(k,t)-robustness, it is clear that if there is no disagreement on predecisions, then rational and correct players are more than $n - t_0$ and thus the protocol terminates and guarantees validity and agreement. If there is instead a disagreement on predecisions then, as long as $m(k,t)$ players commit to bait, by Lemma 4.1 the only outcome is to pay the reward and resolve the disagreement. □

We show in Theorem 4.2 that, provided $m(k,t)$ rational players commit to bait if there is a disagreement in the predecisions, the Trap protocol solves the rational agreement problem. We only have left to prove for which values of $\mathcal{L}$ and $\mathcal{R}$ we can guarantee that the strategy to bait the coalition strictly dominates that of terminating a disagreement for at least $m(k,t)$ rational players in the coalition. We do this in Section 4.3.

4.3 Financial component: deposits & reward

In this section, we focus on the key idea of this paper: what are the values required for a deposit per player and a reward to players for baiting the coalition that make a strong baiting strategy. In particular, and derived from the BFTCR algorithm of Section 4.1, we focus on a baiting strategy that at least $m(k,t)$ rational players will play in Theorem 4.4. Then, we prove that the proposed Trap protocol implements rational agreement and is ϵ-(k,t)-robust for $n > \frac{3}{2}k + 3t$ and $n > 2(k+t)$ in Theorem 4.5 and Corollary 4.1.

We show in Theorem 4.4 which values of $\mathcal{L}$ and $\mathcal{R}$ make the disagreeing strategy a strictly dominated strategy by the baiting strategy for at least $m(k,t)$ rational players (i.e., a dominated strategy even if player i already knows that $m(k,t)-1$ other players are also baiting at the time that i has to decide whether to bait or not). In other words, we show in Theorem 4.4 under which values of $\mathcal{R}$ and $\mathcal{L}$ such strategy $\vec{\eta}$ is a strong $(k,t,m(k,t))$-baiting strategy that satisfies baiting-dominance and lossfree-reward.

The result of Theorem 4.4 is the key part of the Trap protocol for two reasons. First, because it shows that the first $m-1$ baiters do not even prevent a disagremeent from taking place, and thus if the rest of $t+k-(m-1)$ colluding players want to finalize the disagreement, they can. Second, because it shows that if $m-1$ players commit to bait, then the remaining $t+k-(m-1)$ must take the decision on whether to commit to bait or not independently of what the rest of them are doing. Thus, this is analogous to a reduction from the extensive-form game into a normal-form game for this case, played by the $t+k-(m-1)$ remaining rational and Byzantine players, in which all rational players' dominating strategy is to bait the coalition, regardless of what the rest are doing. Without this proof, Byzantine players in the coalition could threat rational players to also bait if they see them baiting, creating a deterrent and changing the equilibrium of rational players into colluding to finalize the disagreement.

We first show in Lemma 4.3 that the Trap protocol guarantees that no player can decide to join the baiting strategy $\vec{\eta}$ and become a valid candidate for the winner consensus after learning that another

$m(k,t)$ players played $\vec{\eta}$: they must take that decision before they know whether $m(k,t)$ other players will play $\vec{\eta}$ or not.

Lemma 4.3. *Let n players play the associated game of the* Trap *protocol $\vec{\sigma}$, out of which k can be rational and t Byzantine, with $n > \frac{3}{2}k + 3t$ and $n > 2(k+t)$. Suppose a run in which a coalition causes a disagreement on predecisions and players start the BFTCR phase. Then, deviating player i in the coalition cannot become a valid candidate for the reward unless it commits to bait before it learns that $m(k,t)$ other players commit to bait.*

Proof. We show that if $m(k,t)$ rational players in the coalition play the baiting strategy, becoming valid candidates to win the reward, then the remaining $k + t - m(k,t)$ cannot obtain valid PoBs to become candidates of the winner consensus after learning that $m(k,t)$ players become candidates. Given that the non-baiting members of the coalition are trying to finalize a disagreement, they will still split non-deviating players into two partitions A and B for the BFTCR protocol. Hence, we look at how many rational players must take part in both partitions of the BFTCR protocol. Notice that $|A|+|B|+t+k \leq n$, $|A|+k+t \geq n-t_0$ and $|B|+k+t \geq n-t_0$. Thus, analogous to how we calculate m in Lemma 3.1, we have that $c \geq (n-t_0) - \frac{n-t-k}{2}$ is the number of members of the coalition that must participate in a partition for it to terminate deciding a predecision, with $A \cap B = \emptyset$, as their predecisions differ. We are interested in calculating $c - t$, the minimum number of rational players out of these c members of the coalition, this is why we include as many Byzantine players as possible. Notice also that we want to see how many rational players must take part in both partitions, meaning that we are interested in $c - t - \frac{k}{2} = (n - t_0) - \frac{n+t}{2} \geq m(k,t)$ for $n > \frac{3}{2}k + 3t$.

Hence, both partitions will include at least $m(k,t)$ repeated rational players. What is left to prove is that if these $m(k,t)$ players commit to bait, then by the time they reveal their commitment, the remaining players cannot collude to try and obtain PoBs to become valid candidates of the winner consensus too. Since $|A|+k+t \geq n-t_0$ and $|B| + k + t \geq n - t_0$, there are $|D| \geq 2(n - t_0) - 2k - 2t$ correct players that delivered at least $m(k,t)$ commitments to bait, for $|D| \leq |A| + |B|$, if these $m(k,t)$ repeated rational players commit to bait. Notice that $|D| \geq t_0 + 1$ for $n > 2(k+t)$. Then, each of the $m(k,t)$ players can wait for $n - t_0$ deliveries of the second reliable broadcast before revealing their commitment by broadcasting their key without compromising termination. Thus, we must calculate for which values of k and t the remaining players cannot obtain PoBs to become valid candidates, that is, for which values of k and t other players that did not bait yet cannot include the new commitment to bait in $t_0 + 1$ valid second reliable broadcasts. Since the remaining set of correct players C such that $|C| = n - t - k - |D|$ are $|C| \geq n - k - t - (2(n - t_0) - 2k - 2t)$, we calculate for which values of k and t we have $|C| + k + t - t_0 \leq t_0$, which results in $n > 2(k+t)$. This means that the $m(k,t)$ baiters can be sure that no deviating player can commit to bait and win the reward without being a valid candidate for the winner consensus. □

We use the result from Lemma 4.3 to prove lossfree-reward and baiting-dominance in Theorem 4.4.

Theorem 4.4 (lossfree-reward and baiting-dominance)**.** *Let $\vec{\sigma}$ be the* Trap *protocol, executed by n players of which exactly k are rational and t Byzantine, for some values of k, t satisfying $n > \max(\frac{3}{2}k + 3t, 2(k+t))$. Let $\vec{\eta}$ be the strategy in which $m(k,t)$ rational players reveal PoFs of the coalition if there is a disagreement on predecisions. Then, $\vec{\eta}$ is a strong baiting strategy if:*

(1) each player is required to deposit $\mathcal{L} = d \cdot \mathcal{G}$, with $d > \frac{m(k,t)}{k(t_0 - m(k,t)+1)}$, and
(2) the baiting reward $\mathcal{R}$ is such that $\mathcal{R} = t_0\mathcal{L}$.

Proof. Recall that the gain is split equally among all k rational players in the coalition $g = \mathcal{G}/k$. To guarantee lossfree-reward, the sum of losses from the coalition must always be equal or greater than the reward given for the coalition to always lose funds while failing to disagree, that is $t_0\mathcal{L} \geq \mathcal{R} \iff \mathcal{L} \geq \frac{\mathcal{R}}{t_0}$.

As a result, the baiting strategy $\vec{\eta}$ must strictly dominate the strategy to disagree for rational players, even if a rational player knows another $m - 1$ other rational players also play the same strategy $\vec{\eta}$ committing to bait. Since the probability of winning the bait between m players is uniformly distributed $p(m) = \frac{1}{m}$ we have that the utility for a player to play the baiting strategy knowing that another $m(k,t) - 1$ players are playing the same strategy is $p(m(k,t))\mathcal{R} - q(m(k,t))\mathcal{L}$. If, instead, the player disagrees then the player's utility is $\frac{\mathcal{G}}{k}$. As such, and since Lemma 4.3 shows that no rational player can become a valid candidate to win the reward after learning that $m(k,t)$ other players commit to bait, we obtain that $\vec{\eta}$ strictly dominates the disagreeing strategy if $p(m(k,t))\mathcal{R} - q(m(k,t))\mathcal{L} > \frac{\mathcal{G}}{k}$ and replacing $\mathcal{R}$ by $t_0\mathcal{L}$, and $\mathcal{L}$ by $d\mathcal{G}$ we obtain:

$$d > \Big(k\big(t_0 p(m(k,t)) - q(m(k,t))\big)\Big)^{-1} \Leftrightarrow \quad d > \frac{m(k,t)}{k(t_0 - m(k,t) + 1)}.$$

As for the reward, $t_0\mathcal{L} \geq \mathcal{R}$ for the slashed deposits to always cover the reward, and thus we set $t_0\mathcal{L} = d\mathcal{G}t_0 = \mathcal{R}$.

Hence, $m(k,t)$ will play the baiting strategy (baiting-dominance) of which one will be rewarded, and the reward will be paid with the deposits of the fraudsters (lossfree-reward). □

Notice that any two values $\mathcal{L}$ and $\mathcal{R}$ suffice if they satisfy $p(m(k,t))\mathcal{R} - q(m(k,t))\mathcal{L} > \frac{\mathcal{G}}{k}$ (1), so that rational players prefer to bait than to disagree, and $t_0\mathcal{L} \geq \mathcal{R}$ (2), so that the reward is always less than the slashed deposits.

The key to these two equations lies in the trade-off between $\mathcal{R}$ and $\mathcal{L}$, that is: $\mathcal{R}$ must be sufficiently big compared to $\mathcal{L}$ so that players prefer to bait than to disagree (Equation 1), but $\mathcal{R}$ must be sufficiently small compared to $\mathcal{L}$ so that the slashed deposits can always pay for the reward (Equation 2).

It is already possible to derive from Theorem 4.4 results for the number of Byzantine players tolerated for $(k - t, t)$-robustness, given a deposit. That is, suppose that $\vec{\eta}$ only requires $m(k,t) = 1$ rational player to satisfy agreement, and let $\mathcal{L} = d \cdot G$, then every coalition of size at least $t_0 + 1$ players has at least $k \geq t_0 + 1 - t$ rational players, and thus the maximum amount of Byzantine players tolerated for ϵ-$(k - t, t)$-robustness is $t < t_0 + 1 - \frac{1}{t_0 d}$. For example, let us set the deposit $\mathcal{L} = d\mathcal{G}$ to $d = \frac{1}{n}$, i.e., the total deposit is $\mathcal{D} = \mathcal{L} \cdot n = \mathcal{G}$, and $n = 100$, it follows that the Trap protocol is ϵ-$(k - t, t)$-robust and $t \leq 30$. If instead $d = \frac{1}{3n}$, then $t \leq 24$.

Finally, we gather all results together in Theorem 4.5, and Corollary 4.1.

Theorem 4.5. *Let $\vec{\sigma}$ be the* Trap *protocol, executed by n players of which k are rational and t Byzantine, for all values k, t satisfying $n > \max(\frac{3}{2}k + 3t, 2(k+t))$. Let $\vec{\eta}$ be the strategy in which $m(k,t)$ rational players reveal PoFs of the coalition if there is a disagreement on predecisions. Then, $\vec{\eta}$ is a strong $(k, t, m(k,t))$-baiting strategy if:*

(1) Each player is required to deposit $\mathcal{L} = d \cdot \mathcal{G}$, where $d > \max_{(k,t)} \left(\frac{m(k,t)}{k(t_0 - m(k,t)+1)}\right)$, and
(2) the baiting reward is $\mathcal{R} = t_0\mathcal{L}$.

Proof. Theorem 4.2 uses the proof of baiting-agreement from Lemma 4.1 to show that if $m(k,t)$ play the baiting strategy in the event of a disagreement on predecisions, then the Trap protocol solves the rational agreement problem. Theorem 4.4 shows that $m(k,t)$ will play the baiting strategy (baiting-dominance) of which one will be rewarded, and the reward will be paid with the deposits of the fraudsters (lossfree-reward).

Finally, we consider all possible values of k and t analogously to Theorem 4.4, deriving a value of d that holds for all possible values of k and t: $d > \max_{(k,t)} \left(\frac{m(k,t)}{k(t_0 - m(k,t)+1)}\right)$. □

Notice that the greater the size of the coalition, the greater d must be in order for the protocol to be ϵ-(k,t)-robust. However, for $n > \frac{3}{2}k + 3t$ and $n > 2(k+t)$, since for every two rational players that join the coalition one Byzantine must leave, the coalition that maximizes the total deposit $\mathcal{D} = \mathcal{L}n = d\mathcal{G}n$ is a coalition of $k = 1$ rational player and $t = t_0$ Byzantine players, and that means $d > \frac{1}{\lceil \frac{n}{3} \rceil - 1}$. Corollary 4.1 shows such particular robustness.

Corollary 4.1. *Let $\vec{\sigma}$ be the* Trap *protocol. Then $\vec{\sigma}$ is ϵ-(k,t)-robust for the rational agreement problem for $n > \frac{3}{2}k + 3t$ and $n > 2(k+t)$ if the following predicates hold:*

(1) Each player is required to deposit $\mathcal{L} = d \cdot \mathcal{G} + \delta$, where $d = \frac{1}{\lceil \frac{n}{3} \rceil - 1}$ and $\delta > 0$, and
(2) the baiting reward is $\mathcal{R} = t_0\mathcal{L}$.

Thus, there are two possible outcomes for the Trap protocol:

- if the coalition is made by so many rational players that deviating does not compensate the risk of losing their deposits, then the Trap protocol will provide agreement at predecision level without paying a reward $\mathcal{R}$, or
- if the coalition has enough Byzantine players to make the deviation into two predecisions profitable, then enough $m(k,t)$ rational players in the coalition will bait so that the disagreement on predecisions can safely be resolved and decided, and one rational player among the baiters will receive a reward $\mathcal{R}$, paid entirely by the deposits of the rest of the provably fraudulent players.

In both scenarios, the Trap protocol implements rational agreement, being ϵ-(k,t)-robust for $n > \max\left(\frac{3}{2}k + 3t, 2(k+t)\right)$.

5 CONCLUSION

In this paper, we showed that rational players help reduce the dependability on correct players to solve the Byzantine consensus problem. To this end, we introduced a necessary and sufficient baiting strategy to solve the rational agreement problem—a variant of the Byzantine consensus problem that also tolerate rational players—under partial synchrony. Based on this strategy, we also proposed Trap, a novel Byzantine consensus protocol among $n > \max\left(\frac{3}{2}k + 3t, 2(k+t)\right)$ players, where k players are rational and t are Byzantine. This protocol tolerates the coordinated deviations of up to k rational players and t Byzantine players, solving consensus in the presence of less than $2n/3$ correct players. As future work, it would be interesting to explore whether our bound $n > \max\left(\frac{3}{2}k + 3t, 2(k+t)\right)$ is tight, and to consider the impact of non-negligible costs of computation and communication. We are currently working at reducing our bound to $n > \frac{3}{2}k + 3t$ with a novel VSS scheme.

Acknowledgements

This research is supported under Australian Research Council Future Fellowship funding scheme (project number 180100496).

REFERENCES

[1] 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*. 501–510.

[2] 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 25th Annual ACM Symposium on Principles of Distributed Computing*. 53–62.

[3] Ittai Abraham, Danny Dolev, and Joseph Y. Halpern. 2019. Distributed Protocols for Leader Election: A Game-Theoretic Perspective. *ACM Transactions on Economics and Computation* 7, 1 (2019).

[4] Ittai Abraham, Philipp Jovanovic, Mary Maller, Sarah Meiklejohn, Gilad Stern, and Alin Tomescu. 2021. Reaching Consensus for Asynchronous Distributed Key Generation. In *Proceedings of the 2021 ACM Symposium on Principles of Distributed Computing*. 363–373.

[5] Yehuda Afek, Yehonatan Ginzberg, Shir Landau Feibish, and Moshe Sulamy. 2014. Distributed Computing Building Blocks for Rational Agents. In *Proceedings of the 2014 ACM Symposium on Principles of Distributed Computing*. 406–415.

[6] Amitanand S. Aiyer, Lorenzo Alvisi, Allen Clement, Mike Dahlin, Jean-Philippe Martin, and Carl Porth. 2005. BAR Fault Tolerance for Cooperative Services. *SIGOPS Operating Systems Review* 39, 5 (2005), 45–58.

[7] Yackolley Amoussou-Guenou, Bruno Biais, Maria Potop-Butucaru, F Paris, and Sara Tucci-Piergiovanni. 2020. Rational vs Byzantine Players in Consensus-based Blockchains. *Proceedings of the 19th International Conference on Autonomous Agents and MultiAgent Systems* (2020), 43–51.

[8] Robert J Aumarm. 2016. Acceptable points in general cooperative n-person games. *Contributions to the Theory of Games (AM-40), Volume IV* 40 (2016), 287.

[9] Xiaohui Bei, Wei Chen, and Jialin Zhang. 2012. *Distributed Consensus Resilient to Both Crash Failures and Strategic Manipulations.* Technical Report 1203.4324. arXiv.

[10] Elchanan Ben-Porath. 2003. Cheap talk in games with incomplete information. *Journal of Economic Theory* 108, 1 (2003), 45–71.

[11] Lorenz Breidenbach, Phil Daian, Florian Tramèr, and Ari Juels. 2018. Enter the Hydra: Towards Principled Bug Bounties and Exploit-Resistant Smart Contracts. In *27th USENIX Security Symposium (USENIX Security 18)*. 1335–1352.

[12] Pierre Civit, Seth Gilbert, and Vincent Gramoli. 2020. Brief Announcement: Polygraph: Accountable Byzantine Agreement. In *Proceedings of the 34th International Symposium on Distributed Computing (DISC) (LIPIcs, Vol. 179)*. 45:1–45:3.

[13] Pierre Civit, Seth Gilbert, and Vincent Gramoli. 2021. Polygraph: Accountable Byzantine Agreement. In *Proceedings of the 2021 IEEE 41st International Conference on Distributed Computing Systems (ICDCS)*. 403–413.

[14] Pierre Civit, Seth Gilbert, Vincent Gramoli, Rachid Guerraoui, and Jovan Komatovic. 2022. As easy as ABC: Optimal (A)ccountable (B)yzantine (C)onsensus is easy!. In *Proceedings of the 36th International Parallel and Distributed Processing Symposium (IPDPS)*.

[15] Tyler Crain, Vincent Gramoli, Mikel Larrea, and Michel Raynal. 2018. DBFT: Efficient Leaderless Byzantine Consensus and its Application to Blockchains. In *2018 IEEE 17th International Symposium on Network Computing and Applications (NCA)*. 1–8.

[16] Varsha Dani, Mahnush Movahedi, Yamel Rodriguez, and Jared Saia. 2011. Scalable rational secret sharing. In *Proceedings of the 30th Annual ACM Symposium on Principles of Distributed Computing*. 187–196.

[17] Sourav Das, Vinith Krishnan, Irene Miriam Isaac, and Ling Ren. 2021. *SPURT: Scalable Distributed Randomness Beacon with Transparent Setup.* Technical Report 2021/100. Cryptology ePrint.

[18] Changyu Dong, Yilei Wang, Amjad Aldweesh, Patrick McCorry, and Aad van Moorsel. 2017. Betrayal, Distrust, and Rationality: Smart Counter-Collusion Contracts for Verifiable Cloud Computing. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 211–227.

[19] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. 1988. Consensus in the presence of partial synchrony. *Journal of the ACM (JACM)* 35, 2 (1988), 288–323.

[20] Zahra Ebrahimi, Bryan Routledge, and Ariel Zetlin-Jones. 2019. *Getting blockchain incentives right.* Technical Report. Carnegie Mellon University.

[21] Michael J Fischer, Nancy A Lynch, and Michael S Paterson. 1985. Impossibility of distributed consensus with one faulty process. *Journal of the ACM (JACM)* 32, 2 (1985), 374–382.

[22] Georg Fuchsbauer, Jonathan Katz, and David Naccache. 2010. Efficient Rational Secret Sharing in Standard Communication Networks. In *Proceedings of the 7th International Conference on Theory of Cryptography (TCC)*. 419–436.

[23] Oded Goldreich, Silvio Micali, and Avi Wigderson. 2019. How to play any mental game. In *Annual ACM Symposium on Theory of Computing*. 307–328.

[24] Adam Groce, Jonathan Katz, Aishwarya Thiruvengadam, and Vassilis Zikas. 2012. Byzantine Agreement with a Rational Adversary. In *Automata, Languages, and Programming*. 561–572.

[25] Adam Groce, Jonathan Katz, Aishwarya Thiruvengadam, and Vassilis Zikas. 2012. Byzantine Agreement with a Rational Adversary. In *Automata, Languages, and Programming*. 561–572.

[26] Joseph Y. Halpern and Xavier Vilaça. 2020. *Rational Consensus.* Technical Report 2005.10141. arXiv.

[27] Itay Harel, Amit Jacob-Fanani, Moshe Sulamy, and Yehuda Afek. 2020. Consensus in Equilibrium: Can One Against All Decide Fairly?. In *23rd International Conference on Principles of Distributed Systems (LIPIcs, Vol. 153)*. 20:1–20:17.

[28] Dominik Harz, Lewis Gudgeon, Arthur Gervais, and William J. Knottenbelt. 2019. Balance: Dynamic Adjustment of Cryptocurrency Deposits. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1485–1502.

[29] Yuval Heller. 2005. *Minority-proof cheap-talk protocol (extended version).* Ph. D. Dissertation. Citeseer.

[30] Eleftherios Kokoris Kogias, Dahlia Malkhi, and Alexander Spiegelman. 2020. Asynchronous Distributed Key Generation for Computationally-Secure Randomness, Consensus, and Threshold Signatures.

[31] A. Kosba, A. Miller, E. Shi, Z. Wen, and C. Papamanthou. 2016. Hawk: The Blockchain Model of Cryptography and Privacy-Preserving Smart Contracts. In *2016 IEEE Symposium on Security and Privacy (SP)*. 839–858.

[32] Leslie Lamport, Robert Shostak, and Marshall Pease. 1982. The Byzantine Generals Problem. *ACM Transaction on Programming Languages and Systems* 4, 3 (1982), 382–401.

[33] Anna Lysyanskaya and Nikos Triandopoulos. 2006. Rationality and Adversarial Behavior in Multi-party Computation. In *Advances in Cryptology - CRYPTO 2006*. 180–197.

[34] Alejandro Ranchal-Pedrosa and Vincent Gramoli. 2021. *ZLB: A Blockchain to Tolerate Colluding Majorities.* Technical Report 2007.10541. arXiv.

[35] Peiyao Sheng, Gerui Wang, Kartik Nayak, Sreeram Kannan, and Pramod Viswanath. 2021. BFT Protocol Forensics. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*. 1722–1743.

[36] Atul Singh, Pedro Fonseca, Petr Kuznetsov, Rodrigo Rodrigues, and Petros Maniatis. 2009. Zeno: Eventually Consistent Byzantine-Fault Tolerance. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*. 169–184.

[37] Ewa Syta, Philipp Jovanovic, Eleftherios Kokoris Kogias, Nicolas Gailly, Linus Gasser, Ismail Khoffi, Michael J. Fischer, and Bryan Ford. 2017. Scalable Bias-Resistant Distributed Randomness. In *2017 IEEE Symposium on Security and Privacy (SP)*. 444–460.

[38] Xavier Vilaça, Oksana Denysyuk, and Luís Rodrigues. 2012. Asynchrony and Collusion in the N-party BAR Transfer Problem. In *Structural Information and Communication Complexity*. 183–194.

[39] 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*.

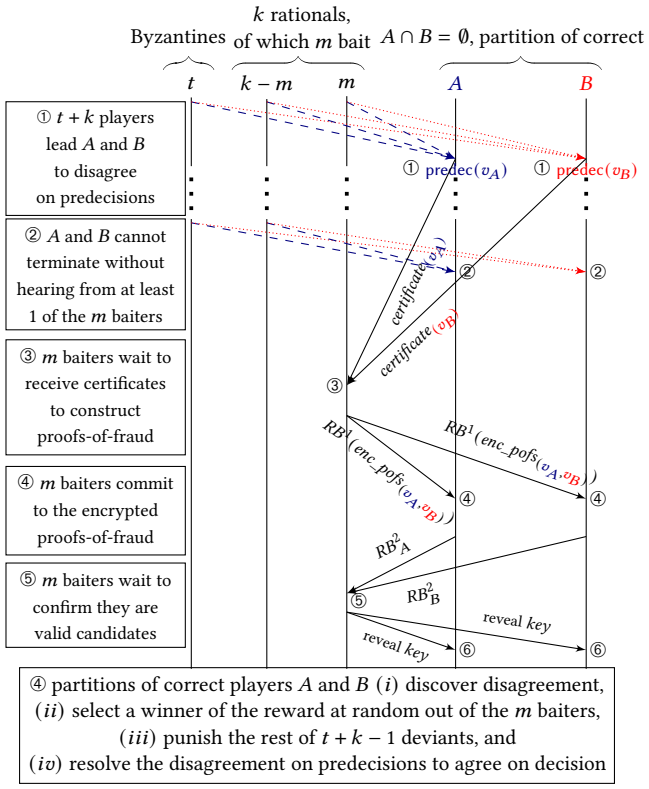

Figure 3: Extended example execution of the TRAP protocol. First, ① all t Byzantine and k rational players collude to cause a disagreement on the output of the accountable consensus protocol, resulting in A and B predeciding different outputs. Then, ② m of the k rational players commit to bait while executing the BFTCR protocol, preventing A and B from deciding their disagreeing predecisions. As such, ③ the m baiters wait until they receive proof of the disagreement on predecisions, to then ④ commit to the encrypted PoFs. Finally, ⑤ once they deliver as many second reliable broadcast from A and B as possible confirming that correct players delivered their PoFs encrypted commitment, then ⑥ the m baiters prove the disagreement revealing the proofs-of-fraud in the BFTCR protocol. Hence, neither A nor B decide their conflicting predecisions, but instead reward one of the m baiters, punish the rest of $t + k - 1$ players responsible for the disagreement on predecisions, and resolve the disagreement, deciding one of v_A or v_B, or, depending on the application, merging both.

A EXTENDED EXAMPLE FIGURE

Figure 3 depicts a slightly extended version of the execution example of Figure 1. Similarly to Figure 1, the execution starts with $k + t$ Byzantine and rational players causing a disagreement on predecisions. However, now we detail further how the m baiters prevent termination of the BFTCR protocol. In particular, by not committing to a value in the first reliable broadcast of BFTCR, the m baiters can prevent players in A and in B from terminating in any of the two partitions. Thus, the m baiting players wait till they receive certificates from players in A and in B in order to construct PoFs. Then, they wait till they deliver enough values from the second group reliable broadcasts from players in partitions A and B that guarantee that no other Byzantine or rational player can become a valid candidate once they reveal that they are baiting (as we showed in the proof of Lemma 4.3). At this point, the m players reveal their PoFs by sending the decryption key to their commitment. Then, players in A and B can resolve their disagreement on predecisions, choose a winner of the reward from among the m valid candidates at random, and punish the rest of deviating players.

B DISCUSSION: PAYING A REWARD AT NO COST TO NON-DEVIANTS.

One might think that implementing a baiting strategy with a reward and deposits might not be enough: we need to discourage coalitions from actually playing the baiting strategy, since the system would have to pay the reward $\mathcal{R}$, and thus the coalition can effectively steal some funds from the system. However, if the system can use the deposited amount $\mathcal{L}$ from at least t_0 certified fraudsters in the coalition to pay for the baiting reward $\mathcal{R}$, then the system does not lose any funds (lossfree-reward), while obtaining agreement (baiting-agreement).

Furthermore, notice that if the coalition consists entirely of rational players then they do not actually play this strong baiting strategy since, by the definition of strong baiting strategy, they all individually lose more than they can gain from deviating. Even if the presence of Byzantine players leads to a baiter being paid, agreement will still be guaranteed at no cost to non-deviating players. This leaves the open question of how likely it is that Byzantine players with unexpected utilities but possibly with the goal to break the system would be interested in giving their funds for free to rational players, if it does not cause some damage on non-deviating players or on the system itself. In other words, with a more refined, realistic modelling of Byzantine players, it is very likely that the very correctness of the TRAP protocol will be enough of a deterrent from deviating, which would lead to agreement directly at the predecision level.