Rational Agreement in the Presence of Crash Faults

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

2021 IEEE International Conference on Blockchain (Blockchain) | 978-1-6654-1760-0/21/$31.00 ©2021 IEEE | DOI: 10.1109/BLOCKCHAIN53845.2021.00072

***Abstract*—Blockchain systems need to solve consensus despite the presence of rational users and failures. The notion of (k,t)-robustness is key to derive impossibility results with k rational players and t faulty players. However, these t faulty players are always considered Byzantine in that they can act arbitrarily. What is less clear is whether these impossibilities hold if these faults are crashes.**

In this paper, we bridge the gap between games that are robust against Byzantine players and games that are robust against crash players. Our first result is an impossibility result: We show that no (k,t)-robust consensus protocols can solve consensus in the crash fault model if $k+2t \geq n$ unless there is a particular punishment strategy, called the (k,t)-baiting strategy. This reveals the need to introduce *baiting* as the act of rewarding a colluding node when betraying its coalition, to make blockchains more secure.

Our second result is an equivalence relation between crash fault tolerant games and Byzantine fault tolerant games, which raises an interesting research question on the power of baiting to solve consensus. To this end, we show, on the one hand, that a (k,t)-robust consensus protocol becomes $(k+t,t)$-robust in the crash model. We show, on the other hand, that the existence of a (k,t)-robust consensus protocol in the crash model that does not make use of a baiting strategy implies the existence of a $(k-t,t)$-robust consensus protocol in the Byzantine model, with the help of cryptography.

***Index Terms*—consensus, game theory, robustness, fault tolerance**

I. Introduction

With the advent of blockchains, there is a growing interest at the frontier between distributed computing and game theory. As one fundamental building block of blockchains is consensus, it is natural to seek equilibria in which consensus is reached despite the presence of both failures and rational players. Moreover, as blockchains handle valuable assets over the Internet, they are typically subject to network attacks [8], [20], [9] and should tolerate unexpected delays—an assumption called partial synchrony [6]—to avoid asset losses.

There are traditionally two types of failures considered in the distributed computing literature: *crash failures* after which a participant stops and *Byzantine failures* when participants act arbitrarily (i.e., irrationally). This is why 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 [19], [11], [5], [2] but rely usually on a trusted central authority, called a *mediator*.

Recent results [1] showed that mediators could be implemented in a fully distributed setting when $n > 3(k+t)$ and t players are Byzantine. Unfortunately, this adaptation makes it impossible to devise even a consensus solution that is immune to a single $t = 1$ Byzantine failure as it is impossible to solve consensus with complete asynchrony and failures [10]. More recent results [12] seem to indicate that when $n \leq 4(k+t)$ there exist equilibria that cannot be implemented in a distributed fashion when the player behaviors are irrational or failures are Byzantine.

These impossibility results raise the question of whether one can solve consensus in a distributed fashion with k rational players and t failures when the failures are crashes. We believe this combination to be particularly relevant in the blockchain context as players are incentivized to steal assets by leading other players to a disagreement—also called a *fork*. Such a situation went undetected in Bitcoin and led the attackers to "double spend", effectively doubling their assets.[1] As blockchains typically run in open networks subject to man-in-the-middle attacks [8], we do not assume synchrony and seek solutions assuming only partial synchrony, where the bound on the message delay is unknown [6].

***Our result**:* In this paper, we focus on designing (k,t)-crash-robust protocols that solve consensus among n players, where k players are rational and up to t players may crash. To this end, we first define (k,t)-*crash-robustness* as an extension of (k,t)-robustness [2] by replacing t Byzantine players by t crash players. As our work is motivated by blockchains where successful attacks against consensus lead players to double spend, we consider that rational players prefer to form a coalition and cause a disagreement than to satisfy agreement. Within this new model, we present robustness bounds and establish constructively a direct relation between (k,t)-robust protocols and (k',t')-crash-robust protocols.

More precisely, we first prove that no resilient-optimal crash-fault tolerant protocol can tolerate even one rational player. Specifically, we prove that it is not possible, in general, to design a protocol that implements consensus and is (k,t)-crash-robust for $k+2t \geq n$, unless there is a (k,t)-crash-baiting strategy (that is, a punishment strategy that strictly dominates deviations towards a disagreement for a number of deviating rational players) with respect to the protocol. This means that state-of-the-art crash-fault tolerant (CFT) protocols

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

978-1-6654-1760-0/21/$31.00 ©2021 IEEE
DOI 10.1109/Blockchain53845.2021.00072

that tolerate up to less than $n/2$ crash players [6] do not tolerate even one rational player in partial synchrony.

This raises a new research question: what would be a protocol that achieves a good compromise between crash-fault tolerance and tolerance to rational players in partial synchrony?

To answer this question, and since the literature tackled the problem for Byzantine faults, we demonstrate a link between Byzantine players with rational and crash players: a protocol that tolerates s Byzantine players also tolerates s rational players and s crash players. We complete this relation by proving that a protocol that implements consensus and is ϵ-(k,t)-robust (where ϵ accounts for the small probability of the adversary breaking the cryptography) is also ϵ-$(k+t,t)$-crash-robust. Additionally, we define a (t',t)-immune protocol as a protocol that tolerates up to t' crash faults and up to t Byzantine faults, and prove that if a protocol is ϵ-(t',t)-immune, then it is also ϵ-$(t,t'+t)$-crash-robust.

Finally, we also establish a relation in the opposite direction assuming cryptography and ignoring crash-baiting strategies: a protocol that tolerates up to k rational players and up to t crash players without a (k,t)-crash-baiting strategy also tolerates up to $\min(k,t)$ Byzantine players, that is, a (k,t)-crash-robust protocol is also $\min(k,t)$-immune. We then prove the relation relative to robustness: if there is a protocol $\vec{\sigma}$ that is ϵ-(k,t)-crash-robust with $k \geq t$ then we can construct an ϵ-$(k-t,t)$-robust protocol $\vec{\sigma'}$, assuming cryptography and that $\vec{\sigma}$ does not implement a (k,t)-crash-baiting strategy. Instead, if $\vec{\sigma}$ implements an effective (k,t)-crash-baiting strategy that is also an effective $(k-t,t)$-baiting strategy, then $\vec{\sigma}$ is also ϵ-$(k-t,t)$-robust protocol, where effective means that playing the baiting strategy still implements consensus. We apply the analogous if $t \geq k$ to obtain that ϵ-(k,t)-crash-robust protocols are also ϵ-$(t-k,k)$-immune, excluding baiting strategies. Finally, we discuss in detail the implications of baiting and punishment strategies to this relation.

A. Related work

Multiple research groups studied the consensus problem by combining rational players with crash faults [4], [15], [3] or by replacing crash faults by rational players [14]. Groce et al. [14] show a protocol that solves Byzantine agreement in synchrony even in the presence of rational coalitions as long as the size k of the coalition is such that $k < n$ and given complete knowledge of the adversary's preferences. Nevertheless, they consider neither crash nor Byzantine (irrational) faults, a model also used by Ebrahimi et al. [7].

Clementi et al. [4] study the problem of fair consensus in the presence of rational and crash faults in the synchronous gossip communication model. Fair consensus adds a new property, *fairness*, defined by all players sharing the same probability of their proposal being decided. The gossip communication model allows all agents to contact at most one neighbor via a push/pull operation at every round.

Harel et al. [17] assume a set of utilities for rational players such that they guarantee solution preference, meaning that all rational players want to satisfy all properties of consensus. Finally, Halpern et al. [15] extend the results on fair consensus in the synchronous model. None of the results listed so far consider either the partially synchronous model or rational players that are interested in disagreeing in blockchains to maximize their profit.

Secret sharing and multi-party computation already explored a combined group of rational players with faulty players [19], [11], [5], [2], specially focused on implementing trusted mediators through cheap-talk [2], i.e., private pairwise communication channels of negligible cost. In particular, Abraham et al. [2] showed in 2006 that there are ϵ-(k,t)-robust protocols that implement mediators with synchronous cheap-talk if $n > k + 2t$, to later extend it in 2019 to ϵ-(k,t)-robust protocols that implement mediators with asynchronous cheap-talk for $n > 3(k+t)$.

Since it is well known that it is impossible to implement a 1-immune protocol that solves consensus in the asynchronous model [18], [10], Ranchal-Pedrosa et al. [21] devised the Huntsman protocol, a protocol that is ϵ-(k,t)-robust and implements the consensus problem with $n > \max(\frac{3}{2}k+3t, 2(k+t))$, which is the highest robustness to date in the presence of Byzantine players and coalitions of rational players that may be interested in causing disagreement. Our result implies that the Huntsman protocol is ϵ-$(k+t,t)$-crash-robust for $n > \max(\frac{3}{2}k+3t, 2(k+t))$.

To the best of our knowledge, we present the first work that obtains bounds for the robustness of agreement against coalitions of crash and rational players in partial synchrony, and that establishes constructively direct relations between (k,t)-robust protocols and (k',t')-crash-robust protocols.

B. Roadmap

The rest of the paper is structured as follows: Section II presents our model and definitions taken from the literature, Section III shows the impossibility of resilient-optimal crash-fault tolerance in the presence of rational players. Section IV establishes the relation between Byzantine players with rational players and crash players, first by showing in Section IV-A that for every Byzantine player tolerated by a consensus protocol, the same protocol can instead tolerate one rational player and one crash player, and second by showing in Section IV-B that if a consensus protocol tolerates k rational players and t crash players then there is a consensus protocol that tolerates $\min(k,t)$ Byzantine players. We finally conclude and detail future work in Section V.

II. Preliminaries

Our focus is on a partially synchronous communication network [6], where messages can be delayed by up to a bound that is unknown, but not indefinitely. For this purpose, we adapt the synchronous and asynchronous models of Abraham et al. [2], [1] to partial synchrony, including the definitions of Halpern et al. [16] to introduce crash players, with the appropriate modifications to account for partial synchrony [21]. Hence, our model consists of a game played by a set N of players,

with $|N| = n$. The players in N can be of four different types: correct, rational, crash or Byzantine.

In order to model partial synchrony, we introduce the scheduler as an additional player that will model the delay on messages derived from partial synchrony. The game is in *extensive form*, described by a game tree whose leaves are labeled by the utility u_i of each player i. 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 targeting i that will be delivered just before i moves (so that i's move can depend on all the messages i receives). Every non-leaf tree 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 up to a bound, known only to the scheduler, before which he must have chosen all receivers of the message 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 the scheduler chooses him 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 tree 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, in that such player cannot distinguish them. We assume that the scheduler has complete information, so that the scheduler's information sets simply consist of singletons.

Since faulty or rational players can decide not to move during their turn, we assume that players that decide not to play will at least play the *default-move*, which consists of notifying to 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 thus alternates with one player at each node down a path in the game tree up to a leaf. A *run* of the game is then a 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, 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 $\mathcal{F}$, 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 desired functionality. Note that both the scheduler and the players can use probabilistic strategies.

Failure model: k players out of n can be rational and up to t of them can be faulty (i.e. Byzantine or crash), while the rest are correct. Correct players follow the protocol: the expected utility of correct player i is equal and positive for any run in which i follows the protocol, and 0 for any other run. Rational players can deviate to follow the strategy that yields them the highest 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). A crash player i behaves exactly as a correct player, except that it can crash in any round of any run. If i crashes in round m of run r, then it may send a message to some subset of agents in round m, but from then on, it sends no further messages (except for playing the default-move). We will detail further the utilities of rational players after defining the Byzantine consensus problem.

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

Cheap talks: As we are in a fully distributed system, without a trusted central entity like a mediator, we assume *cheap-talks*, which are private pairwise communication channels. We also assume negligible communication cost through these channels. Rational and correct players are not interested 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 cryptography, for which we reuse the assumptions of Goldreich et al. [13]: 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.

Robustness: We restate Abraham's et al. [2] definitions of t-immunity, ϵ-(k,t)-robustness and k-resilience. We also add the definitions of t-crash-immune and (k,t)-crash-robust equilibrium. Notice that our definition of (k,t)-crash-robust equilibrium differs from Bei et al's (c,t)-resilient equilibrium, since we simply extend Abraham et al's [2] definition of a (k,t)-robust equilibrium. While this definition is too restrictive for Bei et al's model, it fits properly our illustration of the utilities of the rational players that we define in this work.

A joint strategy $\vec{\sigma}$ 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 faulty players, even if the faulty players join their coalition.

Definition II.1 (ϵ-(k,t)-robust equilibrium)**.** Let K denote the set of $|K| = k$ rational players and let T denote the set of $|T| \leq t$ Byzantine players, $K \cap T = \emptyset$. A joint strategy $\vec{\sigma} \in \mathcal{S}$ is an ϵ-(k,t)-robust equilibrium if for all $K, T \subseteq N$, for all $\vec{\tau}_T \in \mathcal{S}_T$, for all $\vec{\phi}_K \in \mathcal{S}_K$, and all strategies of the scheduler σ_s, there exists $i \in K$ such that:

$$u_i(\vec{\sigma}_{-T}, \vec{\tau}_T, \sigma_s) \geq u_i(\vec{\sigma}_{N-(K \cup T)}, \vec{\phi}_K, \vec{\tau}_T, \sigma_s) - \epsilon.$$

If a k-resilient and t-immune strategy is a (k,t)-robust strategy where $t = 0$ and $k = 0$, respectively. We refer to the technical report for a detailed definition of k-resilience and t-immunity. If $\epsilon = 0$, we simply refer to an ϵ-(k,t)-robust equilibrium as a (k,t)-robust equilibrium, and an ϵ-t-immune protocol as a t-immune protocol.

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 no coalition of size k can gain greater utility by deviating in a coordinated way.

The notion of t-immunity is motivated by the need to tolerate t faulty players. An equilibrium is t-immune if the expected utility of the non-faulty players is not affected by the actions of up to t other faulty players. The ϵ here accounts for the (small) probability of the coalition breaking cryptography, as previously assumed in the literature [2]:

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. We extend this notation to t-immunity and ϵ-(k,t)-robustness. The required functionality of this paper is thus reaching consensus.

Punishment and baiting strategy: We also restate the definition of a punishment strategy [2] as a threat from correct and rational players in order to prevent other rational players from deviating. For example, in a society, this threat can be viewed as a punishment strategy of the judicial system against committing a crime. Slashing a player if he is found to be guilty of deviating is another punishment strategy.

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 II.2 ((k,t)-punishment strategy)**.** Let $K, T, P \subseteq N$ be disjoint sets of rational players, Byzantine players and correct players, respectively, such that $|K| \leq k, |T| \leq t, |P| > t$. A joint strategy $\vec{\rho}$ is a (k,t)-punishment strategy with respect to $\vec{\sigma}$ if 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 $\vec{\rho}$, which decreases the utility of rational players with respect to following the strategy $\vec{\sigma}$.

Recent works consider a particular type of punishment strategies, defined as baiting strategies [21]. In baiting strategies, some players that enforce the punishment on the deviants are rational players from within the coalition: they have an incentive to bait other players into a punishment strategy. An example of a baiting strategy can be found when law-enforcement officers offer an economic reward, or a reduced sentence, if a member of a criminal group helps them arrest the rest of the group.

Definition II.3 ((k,t,m)-baiting strategy)**.** Let $K, T \subseteq N$ be disjoint sets of rational players and Byzantine players, respectively. Let $P \subseteq N$ be a set of baiters, composed of rational and correct players, and let the rest of correct players be $C = N - (K \cup T \cup P)$. 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 $0 < m \leq k$, $|K| \leq k, |T| \leq t, |P| > t, |P \cap K| \geq m$, for all $\vec{\tau} \in \mathcal{S}_T$, all $\vec{\phi}_{K \setminus P} \in \mathcal{S}_{K \setminus P} - \{\vec{\sigma}_K\}$, all $\vec{\theta}_P \in \mathcal{S}_P$, all $i \in P$, and all strategies of the scheduler σ_s, we have:

$$u_i(\vec{\sigma}_C, \vec{\phi}_{K \setminus P}, \vec{\tau}_T, \vec{\eta}_P, \sigma_s) \geq u_i(\vec{\sigma}_C, \vec{\phi}_{K \setminus P}, \vec{\tau}_T, \vec{\theta}_P, \sigma_s)$$

Additionally, we call this strategy a strong (k,t)-*baiting strategy* in the particular case where for all rational coalitions $K \subseteq N$ such that $|K| \leq k + f$, $|K \cap P| \geq m$ and all $\vec{\phi}_{K \setminus P} \in \mathcal{S}_{K \setminus P}$, we have:

$$\sum_{i \in K} u_i(\vec{\sigma}_{N-(K \cup P)}, \vec{\phi}_{K \setminus P}, \vec{\eta}_P, \sigma_s) \leq \sum_{i \in K} u_i(\vec{\sigma}, \sigma_s).$$

We write (strong) (k,t)-baiting strategy instead to refer to a (strong) (k,t,m)-baiting strategy for some m, with $0 < m \leq k$.

We also refer to an *effective* baiting strategy if playing such strategy still implements the desired functionality. An example of an effective baiting strategy for the problem of consensus is rewarding a baiter for exposing a disagreement attempt before it takes place, if it is resolved [21]. Strong baiting strategies are strategies in which the coalitions formed entirely by rational players end up collectively losing compared to if they had just followed the protocol, even if a subset of them play the baiting strategy. This prevents coalitions from baiting themselves just for the purpose of maximizing the sum of their utilities.

We extend the above-defined terms to their analogous crash fault tolerant counterparts by replacing Byzantine players by crash players in all their definitions, in what we refer to as (t)-crash-immunity, (k,t)-crash-robustness, (k,t)-crash-punishment strategy and (k,t,m)-crash-baiting strategy.

Consensus: We recall the Byzantine consensus problem [18] in the presence of rational players: The *Byzantine consensus problem* is, given n players, each with an initial value, to ensure (1) *Agreement*, in that no two non-deviating players decide different values, (2) *Validity* in that if a non-deviating player decides a value v, then v has been proposed

by some player, and (3) *Termination* in that all non-deviating players eventually decide.

Disagreements: Notice a disagreement of consensus can mean two or more disjoint groups of non-deviating players deciding two or more conflicting decisions [23]. We speak of the *disagreeing* strategy as the strategy in which deviating players collude to produce a disagreement, and of a coalition *disagreeing* to refer to a coalition that plays the disagreeing strategy.

Rational players: The utilities of a rational player i depending on its actions and a particular run of the protocol are as follows:

1) If an agreement is reached, then i gets utility $u_i(\vec{\sigma}_{-T}, \vec{\tau}_T) \geq \epsilon$ where $\epsilon > 0$.
2) If coalition where i is a member successfully performs a disagreement, then i gets utility $u_i(\vec{\sigma}_{N-K-T}, \vec{\phi}_{K \cup T}) = g > \epsilon$.
3) If the protocol does not terminate, then i obtains a negative utility.
4) If player i suffers a disagreement caused by a coalition external to i, then i obtains a negative utility.

Note that we do not consider fairness of consensus, computational costs, communication costs, or a preference from a proposal over another, but we rather focus on the interests in causing a disagreement. As such, setting the expected utility of causing agreement to be greater than that of causing disagreement would inevitably lead to rational players behaving exactly as correct players. Similarly, selecting the utilities of not terminating greater than the utilities of causing agreement would lead to all rational players behaving as crash players. For both of these cases, the state-of-the-art bounds are applicable. The set of utilities that we choose here also reflects a realistic behavior of rational players in the blockchain context, where players can get an economic incentive from a fork (as it is the case when they double spend by forking). It is easy to notice that not terminating as well as deciding a proposal while a coalition causes a disagreement are strictly dominated by reaching agreement, which is in turn strictly dominated by causing a disagreement. A baiting strategy means a rational player possibly joining the coalition only to later betray it in exchange of a reward. This additional strategy would strictly dominate the strategy to disagree by definition.

III. Impossibility result

In this section, we show that resilient-optimal protocols cannot tolerate even one rational player. We refer to the technical report for the proofs [22]. Previous results showed that a resilient-optimal protocol tolerates up to $t < n/2$ crash faults [6]. We show in Theorem III.1

Theorem III.1. *Let $\vec{\sigma}$ be a protocol that implements consensus such that there is no (k,t)-crash-baiting strategy with respect to $\vec{\sigma}$. Then, it is impossible for $\vec{\sigma}$ to be (k,t)-crash-robust for $k + 2t \geq n$.*

Theorem III.1 shows that it is necessary to consider new bounds for CFT protocols in terms of their crash-fault tolerance, since their resilient-optimal bounds make them vulnerable to even one rational player for state-of-the-art protocols, because they do not consider offering a crash-baiting strategy. In Section IV, we explore the link between crash-robustness and immunity, so as to obtain results for this model with the existing protocols designed for Byzantine faults.

IV. Bridging the gap: crash and rational players as Byzantine players

In this section we bridge the gap between games that are robust against Byzantine players and games that are robust against crash players.

A. From Byzantine to crash players

It is immediate that a protocol that tolerates up to t Byzantine faults also tolerates up to t crash faults, the question lies with the inclusion of rational players. We propose in Theorem IV.1 a first relation between Byzantine fault tolerance and crash-fault tolerance in the presence of rational players.

Notice that the statement of Theorem IV.1 does not require to assume cryptography, and thus the same result takes place by considering $\epsilon = 0$, i.e., (k,t)-robustness. Theorem IV.1 establishes a surprising yet meaningful relation between t-immunity and (k,t)-crash-robustness: if a protocol is (k,t)-robust then it is also $(k+t,t)$-crash-robust.

Theorem IV.1. *Let $\vec{\sigma}$ be a protocol that implements consensus and is ϵ-(k,t)-robust. Then, $\vec{\sigma}$ is also ϵ-$(k+t,t)$-crash-robust.*

By Theorem IV.1, it is possible to take existing protocols, bounds and other results that apply to immunity and robustness and apply them directly to crash-immunity and crash-robustness.

Interestingly, an analogous proof provides the same result for a protocol that tolerates instead crash and Byzantine players. We show this result in Theorem IV.2. However, we first define ϵ-(t',t)-immunity to combine tolerance to a number of crash and Byzantine players together:

Definition IV.1 (ϵ-(t',t)-immunity)**.** A joint strategy $\vec{\sigma} \in \mathcal{S}$ is ϵ-(t,t')-immune if, for all sets T of Byzantine players such that $T \subseteq N$ with $|T| \leq t$, all sets T' of crash players such that $T' \subseteq N$, $T \cap T' = \emptyset$, all $\vec{\tau} \in \mathcal{S}_T$, all $\vec{\theta} \in \mathcal{S}_{T'}$, all strategies σ_s of the scheduler, and all $i \notin T \cup T'$, we have:

$$u_i(\vec{\sigma}_{-\{T \cup T'\}}, \vec{\tau}_T, \vec{\theta}_{T'}, \sigma_s) \geq u_i(\vec{\sigma}, \sigma_s) - \epsilon.$$

Theorem IV.2. *Let $\vec{\sigma}$ be a protocol that implements consensus and is ϵ-(t',t)-immune. Then, $\vec{\sigma}$ is also ϵ-$(t,t'+t)$-crash-robust.*

B. From crash to Byzantine players

One may wonder if the same result listed in Theorem IV.1 is true in the opposite direction, that is, whether a protocol $\vec{\sigma}$ that is ϵ-(k,t)-crash-robust is also ϵ-$(k-t,t)$-robust. We show this result in Theorem IV.3.

Theorem IV.3. *Let $\vec{\sigma}$ be an ϵ-(k,t)-crash-robust protocol that implements consensus without a (k,t)-crash-baiting strategy*

with respect to $\vec{\sigma}$, where $k \geq t$. Then, assuming cryptography and a public-key infrastructure scheme, there is an ϵ-$(k-t,t)$-robust protocol $\vec{\sigma}'$ that implements consensus.

Theorem IV.3 excludes protocols that make use of a (k,t)-crash-baiting strategy $\vec{\eta}$ to be ϵ-(k,t)-crash-robust, we show in Theorem IV.4 that if $\vec{\eta}$ is both an efficient (k,t)-crash-baiting strategy and an efficient (k,t)-crash-baiting strategy, then there is a protocol that is ϵ-$(k-t,t)$-robust.

Theorem IV.4. *Let $\vec{\sigma}$ be an ϵ-(k,t)-crash-robust protocol that implements consensus such that there is an efficient (k,t)-crash-baiting strategy $\vec{\eta}$ with respect to $\vec{\sigma}$, where $k \geq t$. Let $\vec{\sigma'}$ be the BFT-extension of $\vec{\sigma}$, assuming cryptography and a public-key infrastructure scheme. If there is $\vec{\eta'}$ such that $\vec{\eta'}$ is also an efficient $(k-t,t)$-baiting strategy with respect to $\vec{\sigma'}$, then $\vec{\sigma'}$ is an ϵ-$(k-t,t)$-robust protocol that implements consensus.*

Again, notice that the results from Theorems IV.3 and IV.4 assume $k \geq t$, if instead $t \geq k$, then we obtain the result from theorems IV.5, for which we reuse the definition of ϵ-(t,t')-immunity from Section IV-A.

Theorem IV.5. *Let $\vec{\sigma}$ be an ϵ-(k,t)-crash-robust protocol that implements consensus without a (k,t)-crash-baiting strategy with respect to $\vec{\sigma}$, where $t \geq k$. Then, assuming cryptography and a public-key infrastructure scheme, there is an ϵ-$(t-k,k)$-immune protocol $\vec{\sigma}'$ that implements consensus.*

V. Conclusion & Future Work

In this paper, we showed two interesting relations between types of faults in the solvability of the rational agreement. First, if a consensus protocol is ϵ-(k,t)-robust then it is also ϵ-$(k+t,t)$-crash-robust, meaning that in the absence of irrational (Byzantine) players, one can tolerate coalitions with t additional rational players. Second, with cryptography but in the absence of a baiting strategy then we can devise a ϵ-$(k-t,t)$-robust consensus protocol from a ϵ-(k,t)-crash-robust consensus protocol. We also prove that if a protocol is ϵ-(t',t)-immune, then it is also ϵ-$(t,t'+t)$-crash-robust, and that if a protocol $\vec{\sigma}$ is ϵ-(k,t)-crash-robust, where $t \geq k$, then there is an ϵ-$(t-k,k)$-immune protocol $\vec{\sigma}'$ that implements consensus, excluding baiting strategies. We can conclude, thanks to the results here outlined, that the Huntsman protocol [21] yields the greatest crash-robustness to date under this model, as it is $(k+t,t)$-crash-robust for $n > \max(\frac{3}{2}k+3t, 2(k+t))$.

References

[1] Ittai Abraham, Danny Dolev, Ivan Geffner, and Joseph Y. Halpern. Implementing mediators with asynchronous cheap talk. In *Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing*, PODC '19, page 501–510, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3293611.3331623.

[2] Ittai Abraham, Danny Dolev, Rica Gonen, and Joe Halpern. Distributed computing meets game theory: Robust mechanisms for rational secret sharing and multiparty computation. In *PODC*, pages 53–62, 2006.

[3] Xiaohui Bei, Wei Chen, and Jialin Zhang. Distributed consensus resilient to both crash failures and strategic manipulations. *arXiv preprint arXiv:1203.4324*, 2012.

[4] A. Clementi, L. Gualà, G. Proietti, and G. Scornavacca. Rational fair consensus in the gossip model. In *2017 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*, pages 163–171, 2017. doi:10.1109/IPDPS.2017.67.

[5] Varsha Dani, Mahnush Movahedi, Yamel Rodriguez, and Jared Saia. Scalable rational secret sharing. In *PODC*, pages 187–196, 2011.

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

[7] Zahra Ebrahimi, Bryan Routledge, and Ariel Zetlin-Jones. Getting blockchain incentives right. Technical report, Tech. rep., Carnegie Mellon University Working Paper, 2019.

[8] Parinya Ekparinya, Vincent Gramoli, and Guillaume Jourjon. Impact of man-in-the-middle attacks on ethereum. In *2018 IEEE 37th Symposium on Reliable Distributed Systems (SRDS)*, pages 11–20, 2018. doi:10.1109/SRDS.2018.00012.

[9] Parinya Ekparinya, Vincent Gramoli, and Guillaume Jourjon. The attack of the clones against proof-of-authority. *NDSS Symposium*, 2020.

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

[11] Georg Fuchsbauer, Jonathan Katz, and David Naccache. Efficient rational secret sharing in standard communication networks. In *Proceedings of the 7th International Conference on Theory of Cryptography (TCC)*, page 419?436, 2010.

[12] Ivan Geffner and Joseph Y. Halpern. Lower bounds implementing mediators in asynchronous systems. *arXiv preprint arXiv:2104.02759*, 2021.

[13] Oded Goldreich, Silvio Micali, and Avi Wigderson. How to play any mental game. In *Annual ACM Symposium on Theory of Computing*, 1987.

[14] Adam Groce, Jonathan Katz, Aishwarya Thiruvengadam, and Vassilis Zikas. Byzantine agreement with a rational adversary. In *International Colloquium on Automata, Languages, and Programming*, pages 561–572. Springer, 2012.

[15] Joseph Y. Halpern and Xavier Vilaça. Rational consensus: Extended abstract. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, page 137–146, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2933057.2933088.

[16] Joseph Y. Halpern and Xavier Vilaça. Rational consensus: Extended abstract. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, page 137–146, New York, NY, USA, 2016. Association for Computing Machinery. doi:10.1145/2933057.2933088.

[17] Itay Harel, Amit Jacob-Fanani, Moshe Sulamy, and Yehuda Afek. Consensus in Equilibrium: Can One Against All Decide Fairly? In Pascal Felber, Roy Friedman, Seth Gilbert, and Avery Miller, editors, *23rd International Conference on Principles of Distributed Systems (OPODIS 2019)*, volume 153 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:17, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2020/11806, doi:10.4230/LIPIcs.OPODIS.2019.20.

[18] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Trans. Program. Lang. Syst.*, 4(3):382–401, July 1982.

[19] Anna Lysyanskaya and Nikos Triandopoulos. Rationality and adversarial behavior in multi-party computation. In Cynthia Dwork, editor, *CRYPTO*, pages 180–197, 2006.

[20] Christopher Natoli and Vincent Gramoli. The balance attack or why forkable blockchains are ill-suited for consortium. In *2017 47th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 579–590, 2017. doi:10.1109/DSN.2017.44.

[21] Alejandro Ranchal-Pedrosa and Vincent Gramoli. Agreement in the presence of disagreeing rational players: The huntsman protocol. *arXiv preprint arXiv:2105.04357*, 2021.

[22] Alejandro Ranchal-Pedrosa and Vincent Gramoli. Rational agreement in the presence of crash faults. *arXiv preprint arXiv:2111.01425*, 2021.

[23] Atul Singh, Pedro Fonseca, Petr Kuznetsov, Rodrigo Rodrigues, and Petros Maniatis. Zeno: Eventually consistent byzantine-fault tolerance. In *Proceedings of the 6th USENIX Symposium on Networked Systems Design and Implementation*, NSDI'09, page 169–184, USA, 2009. USENIX Association.