Rebop: Reputation-Based Incentives in Committee-Based Blockchains

Arian Baloochestani, Leander Jehl, Hein Meling

Rebop: Reputation-Based Incentives in Committee-Based Blockchains

Arian Baloochestani(✉), Leander Jehl, and Hein Meling

Department of Electrical Engineering and Computer Science, University of Stavanger, Stavanger, Norway
{arian.masoudbaloochestani,leander.jehl,hein.meling}@uis.no

Abstract. Blockchains based on proof-of-work suffer from serious drawbacks, such as high computational overhead, long confirmation time, and forks. Committee-based blockchains provide an alternative that tackles these problems. These blockchains use a committee to approve a block at each height. However, rewarding the committee for their work is challenging. The reward mechanism must be fair and robust to attacks.

In this paper, we study leader-based reward mechanisms in committee-based blockchains in the presence of rational, colluding, and Byzantine committee members. First, we study the incentives of committee members to deviate and show that an existing reward mechanism is susceptible to attacks from both colluding and Byzantine members.

We then propose a reputation-based leader selection mechanism that provides sufficient incentives to coerce rational members to abide by the protocol, and significantly limits the possible gains of collusion. Additionally, our approach reduces the ability of Byzantine members to perform targeted attacks.

Keywords: Committee-based blockchains · Reward mechanisms · Incentives · Reputation-based rewarding · Fairness

1 Introduction

The blockchain was first introduced in 2008 as an infrastructure for the Bitcoin cryptocurrency [27] and has since become an appealing technology for various applications. A *blockchain* is a secure database where users share their data in a distributed and trusted environment [34]. The unknown and untrusted participants that maintain a blockchain do not rely on a trusted third party [15].

The foundation of a blockchain is its underlying consensus protocol. Processes acting on behalf of users produce blocks of transactions, and consensus protocols

This work is partially funded by the BBChain and Credence projects under grants 274451 and 288126 from the Research Council of Norway.

Published by Springer Nature Switzerland AG 2022
D. Eyers and S. Voulgaris (Eds.): DAIS 2022, LNCS 13272, pp. 37–54, 2022.
https://doi.org/10.1007/978-3-031-16092-9_4

determine how *participating processes* agree on which block to append next to the blockchain [5]. This allows processes to securely and consistently update shared states following the state machine approach [33].

There are various kinds of consensus protocols with different configurations and characteristics. In Proof-of-Work (PoW) [27] and Proof-of-Stake (PoS) [32], a single participating process is selected to propose a new block and successively rewarded if the block was valid. The probability of selecting a process is proportional to the energy and computational resources spent in PoW, or the amount of digital currency the process has invested in PoS. While PoS avoids the tremendous amounts of resources used by PoW blockchains, the mechanism suffers from various security problems such as the *nothing at stake* and the *long-range attacks* [21]. Therefore, some blockchains use a combination of PoS and a committee to overcome these drawbacks.

In committee-based blockchains, a group of processes is responsible for updating the blockchain. Numerous committee-based blockchains exists, such as Tendermint [17], LibraBFT/HotStuff [6,37], Algorand [8], and HyperLedger Fabric [4]. In these blockchains, one process is selected as the leader to propose a new block. The other committee members (aka validators) vote for this block. If a majority of validators vote for the block, it will be added to the blockchain.

Shifting the responsibility for block creation from a single process to a committee requires adjusting the reward mechanism. A fair reward mechanism should reward participating committee members and prevent free-riding processes from gaining rewards [3,25]. Designing such mechanisms involves multiple tradeoffs. The key challenges include tolerating message loss and transient outages of individual processes. Repeated retransmissions and reconfiguration can address these challenges but requires complex protocol adaptations [20]. Leader-based mechanisms are more efficient but suffer from false detections, which both benign and malicious leaders may trigger.

This paper analyzes leader-based reward mechanisms and their robustness against different attacks. Unlike previous work, we consider misbehavior from rational, colluding, and Byzantine committee members. Rational and colluding attackers try to increase their share of rewards and can be dissuaded by proper incentives. Byzantine members, however, may perform attacks regardless of the offered incentives, e.g., motivated by factors outside the system. Further, such attacks may target individual members instead of the system as a whole.

We propose Reputation-based Reward Opportunity (Rebop), which relies on reputation-based leader election to give well-behaved processes opportunities to earn a bonus for serving as leaders. Taking longer behaviour into account, Rebop is able to distinguish between a constant and one-time misbehaviour and thus significantly reduce the profitability of attacks. Different from pure monetary mechanisms, reputation-based leader election can also reduce the capabilities of Byzantine attackers, that may not care about lost rewards.

We devise a normal form game-theoretic framework for incentive schemes to determine their robustness against attacks from rational and colluding committee members. We model Rebop and Cosmos' incentive scheme [18] in this

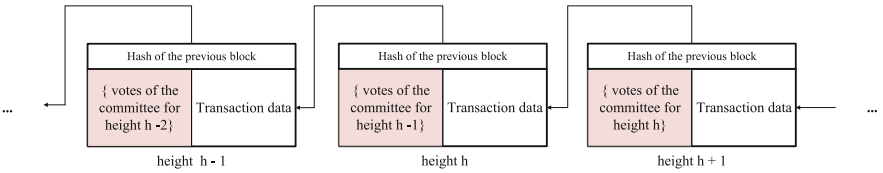

Fig. 1. Blockchain structure. Each block contains data, the previous block's hash, and proof of commit. The proof contains votes from the committee for the previous block.

framework. Our analysis shows that Rebop and Cosmos require similar bonuses to thwart attacks up to a given coalition size. However, for larger coalitions, profitable misbehavior is significantly restricted in Rebop compared to Cosmos. Further, Cosmos provides no countermeasures to restrict Byzantine behaviors.

We use simulations to verify our analytical results and evaluate our reputation-based method in more complex scenarios, including multiple concurrent attacks and message loss.

2 Committee-Based Blockchains

A blockchain is stored as a cryptographically secured append-only log that is shared among several processes. Each block or entry in the log contains data; for example, in cryptocurrencies like Bitcoin, this data is a set of new transactions in which money is transferred from one user to another. Additionally, every block contains a cryptographic hash of its predecessor, as shown in Fig. 1. These hashes ensure the integrity of the stored data. Users of the system are identified by a public key and authorized through digital signatures [22]. To ensure a consistent system state, i.e., account balances, processes need to agree on the order in which blocks are appended to the blockchain and transactions are executed. This is achieved through a consensus algorithm. The number of blocks between the genesis block and a particular block is called the *block height*.

In some consensus algorithms, such as PoW, processes compete to find and issue a new block; thus, different processes may produce more than one valid block at a particular height. This leads to different paths in the blockchain called forks, and consequently, processes will be confused about which fork to follow. To prevent forks, some blockchains use a committee to confirm the new block proposed by a leader [6,8,9,13,17]. In these blockchains, at every height, a leader is elected, responsible for proposing a new block. Then, other committee members vote for the proposed block if it is valid. The block is committed if a sufficiently large fraction of the members vote for the block in one or more rounds. The fraction and the number of rounds depend on the algorithm.

Different committee-based blockchains employ public or private leader election procedures. In private leader election, processes can secretly determine if they are the leader and publish proof of such leadership. Some blockchains, such as Algorand [8] and Snow White [9], use verifiable random functions [24] to

produce uniformly distributed random values with non-interactive proofs. All processes run the function privately at every height, and its output determines the leader. The selected leader can present proof of leadership along with the proposed block to any process. In Algorand, committee members and leaders are chosen randomly with probabilities proportional to their stakes, and more than one leader may get elected for each round.

In a public leader election, all processes can infer who will be the next leader. Typically, the next leader depends on randomness derived from the previous round. In Dfinity [13], this randomness is the input of a pseudo-random permutation. The original Tendermint [17] protocol uses round-robin for electing the leader in each round. However, in current Tendermint, referred to as Cosmos [18], the probability of becoming a leader is proportional to the processes' stake.

3 System and Protocol Model

In this section, we discuss the system model and the related assumptions. In addition, we give a high-level model for a committee-based blockchains that suits multiple protocols.

We assume a set $\Pi = \{p_1, p_2, ..., p_n\}$ of processes which are all functioning as committee members. This assumption fits well for consortium-based or permissioned blockchains. However, PoS-based blockchains may also exhibit a relatively stable committee. For example, in Cosmos, 125 processes with the most stake are selected in the committee, and they remain until replaced by other processes with more stake. We note that it is common to pose restrictions on how quickly deposited stake may be withdrawn [28]. Moreover, while our methods focus on the leader, they could also be applied to systems that randomly select the committee from a larger set of processes. We assume that the network is synchronous, but it may lose messages.

We assume that every process p_i has a voting power $m_i \in (0, 1)$, such that $\sum_{p_i \in \Pi} m_i = 1$. Typically, voting power will be evenly distributed among processes. To model coalitions, we also allow a process to control a larger fraction of the total voting power than its fair share.

In our *blockchain model* progress is measured through a parameter height h, which represents the current length of the blockchain. A block is added to the chain at each height, following the process in Fig. 2. The details of individual consensus algorithms are abstracted.

Several rounds might be needed for a block to be approved at some height h. At each round t, a leader $L_t \in \Pi$ is selected to propose a new block. We assume that leader selection is randomized and write $P[L_t = p_i]$ for the probability that p_i becomes the leader at round t. We further assume that $P[L_t = p_i]$ may depend on the voting power m_i and the history of the blockchain up to height $h - 1$.

To publish a new block at height h, the leader needs to collect votes for the previous block, proposed at height $h - 1$. As shown in Fig. 1, these votes need to be included in the new block. We use a parameter f to specify, the amount of voting power for which votes may be missing:

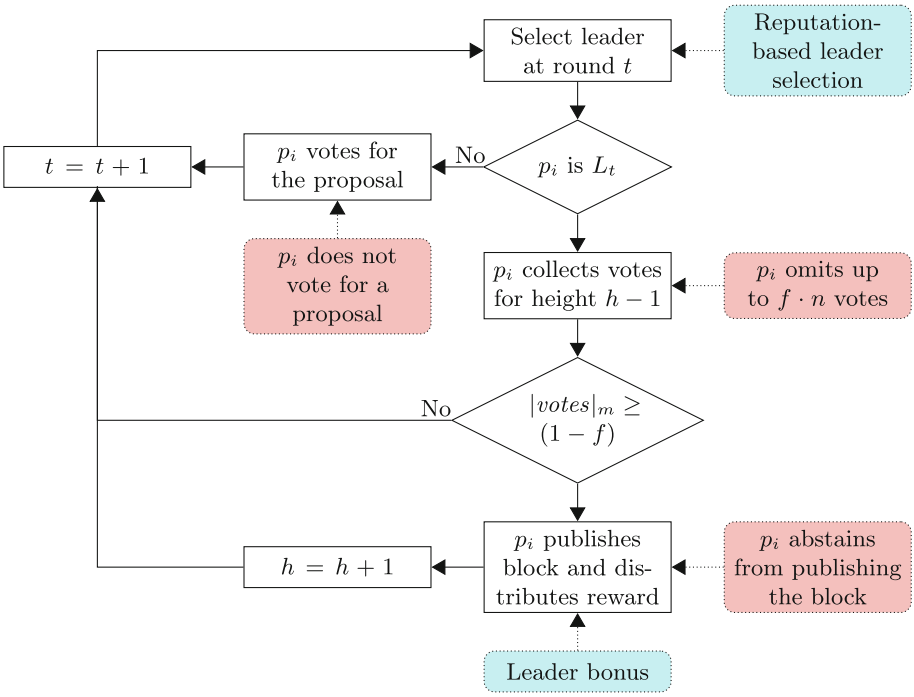

Fig. 2. Overview of the system model. At every height, a leader collects votes for the preceding block and publishes a new block. The figure also shows possible attacks (red) and countermeasures (blue) discussed in Sects. 4 and 5. (Color figure online)

$$|votes|_m = \sum_{p_i \in voted} m_i \geq 1 - f, \quad \text{where } f \text{ is typically } \frac{1}{3},$$

If the leader cannot collect enough votes the system progresses to the next round. The parameter f typically also specifies the amount of voting power (processes) that can be faulty, and may vary depending on the protocol. Committee members sign their votes with their private keys; hence, the identity of each vote is known to all processes. We assume that digital signatures cannot be forged. Processes broadcast their votes for the next potential leader to collect.

A reward R is paid out to motivate the processes to follow the protocol at height h. The reward is distributed, according to their voting power, among the processes whose votes are included at height $h + 1$. This ensures that processes that did not participate do not receive a reward. We assume that R is constant and is not related to the contents of the block. R may be fixed in a cryptocurrency due to economic concerns such as inflation and money circulation.

As shown in Fig. 2, a faulty leader may omit some votes. Next, we discuss how and why this might happen and how it affects the utility of the processes.

4 Attacks and Incentives

Some processes can exhibit malicious behavior by deviating from the protocol. More precisely, the model presented in Sect. 3 allows three different attacks:

I A member of the committee may not vote for a proposal or fail to disseminate the proposal.
II A leader may abstain from publishing a block.
III A leader may omit some of the votes when publishing a block.

We follow the BAR model [1], assuming altruistic, rational, and Byzantine processes. Additionally, in a system with open membership, a single entity may control multiple processes. Therefore, we also consider the possibility of colluding rational processes [35].

Altruistic. Altruistic or correct processes strictly follow the protocol. Correct processes may experience network failures, and their messages may get lost.

Rational. Rational processes follow the protocol unless deviating increases the reward. These processes vote to get the reward, abstaining from Attack I.

However, a rational leader may exclude some votes to increase the share of the reward received for his own vote (Attack III). If a rational process p_r with voting power m_r voted at height h and is selected as the next leader $p_r = L_{h+1}$, then it may increase its share of the reward by omitting a fraction $e < f$ of the votes. Thus, instead of the honest share $share[\text{honest}] = m_r R$, p_r will receive $share[\text{omit}_e] = \frac{m_r R}{(1-e)}$. We can see that p_r has a solid motivation to deviate from the protocol to maximize its share of the reward. Authors in [17] claim that this problem will not occur due to the tit-for-tat strategy taken by the validators; however, due to the probability of message loss in the network, no one can prove that it is excluded from the reward intentionally. Thus, a process that is subject to message loss would suffer unfairly from such retaliation.

Finally, resource constraint rational processes may also try to avoid the additional steps performed by a leader, leading to Attack II.

Colluding. While a rational process p_r deviates from the protocol if that leads to more profit, a coalition works together to increase the group's total profit. We model colluding processes as a single process with a larger voting share. Similar to rational processes, colluding processes also have the same motivation to perform Attack III.

Byzantine. Byzantine processes arbitrarily deviate from the protocol. Unlike rational processes, Byzantine attackers do not care about their outcome; because they have an external motivation unknown to anyone else. They may, for example, try to harm the system or specific other processes. Committee based protocols remain functional despite a certain fraction (f) of Byzantine processes. We therefore ignore attacks on the protocol in this work and focus on the rewarding mechanism, especially on targeted attacks, where Byzantine processes try to

reduce the reward of targeted committee members. In such a targeted attack, a Byzantine process may selectively distribute its vote (Attack I) or, if selected as the leader, ignore the votes of some processes in the committee (Attack III). Note that incentives cannot discourage Byzantine behavior because Byzantine processes are motivated by external goals. Instead of monetary punishment, we need to reduce Byzantine processes' ability to conduct attacks.

In the next section we present our reputation-based incentive scheme.

5 Rebop: Reputation-Based Reward Opportunity

In a committee-based scheme, the leader role carries a special responsibility and must perform additional tasks. Therefore, the leader should be rewarded more than other committee members. Additionally, this reward should discourage rational or colluding processes from omitting votes. We note that benign leaders may also lose votes due to message loss. In our incentive scheme, Rebop, we reward correct leaders with the possibility of additional earnings rather than punishing misbehavior.

As the flowchart in Fig. 2 indicates, Rebop combines two mechanisms. We use reputation-based leader election to select leaders in each round. In addition, we propose to give a fixed fraction of the block reward as a bonus to the leader to enforce long-term benefits for rational and colluding processes. The bonus encourages leaders to actually propose a block, preventing Attack II.

If we penalize deviating processes for Attack III by selecting them less often as the leader in the subsequent blocks, we reduce the ability of Byzantine attackers. Additionally, deviating processes are punished by losing bonus now given to other leaders. This can motivate rational processes against Attack III. Rebop computes *reputation* based on the average number of votes a process p_i has included as the leader during the last T blocks. Let $leader(i, h)$ determine whether p_i was a leader at height h:

$$leader(i,h) = \begin{cases} 1 & \text{if } p_i \text{ is } L_h \\ 0 & \text{otherwise} \end{cases}$$

Then, the reputation $r_{i,h} \in [0, 1]$ of p_i at height h is calculated as:

$$r_{i,h} = \begin{cases} 1 & \text{if } \sum_{t=h-T}^{h} leader(i,t) = 0 \\ \frac{\sum_{t=h-T}^{h} leader(i,t) \cdot \left(\frac{f-e_t}{f}\right)^{\alpha}}{\sum_{t=h-T}^{h} leader(i,t)} & \text{otherwise} \end{cases} \quad (1)$$

where $e_t \in [0, f]$ is the number of votes missing from the block at height t, and $\alpha \geq 1$ is a parameter of the protocol. T should be selected in a way to allow each process to become the leader at least once during the next T rounds. For $\alpha > 1$ repeated omission of even a few votes results in a lower reputation than a one time omission of many votes. This helps to reduce the ability of Byzantine attackers to omit individual players. Thus, larger α gives better protection from Byzantine attackers, but may open to additional attacks from colluding processes

as we show below. A large α may also result in punishments for correct processes that suffer from message loss.

We write r_i for the reputation of p_i at the current height The chances of the process p_i to be selected as the leader is proportional to r_i and m_i. The more the reputation of p_i, the more chances for it to be the leader, and consequently, the more bonus it gets.

$$P[p_i = L_h] = \frac{r_i m_i}{\sum_{i \leq n} r_i m_i} \tag{2}$$

Rational players may not produce a block if they lose too many votes to prevent their reputation from being slashed (Attack II). However, because reputation is an average of the number of votes gathered in the last T blocks, a small value for reputation in one round cannot affect the total reputation much.

In addressing Attack III, we note that reputation-based leader election can make Attack I more attractive for rational and colluding players. By omitting votes and reducing the reputation of other processes, rational processes may try to gain a larger share of the rewards. However, our analysis in the next section shows that the reward lost to this attack is often higher than the earned bonuses.

6 Incentive Analysis

We use game theory to analyze the different strategies of committee members. Specifically, we use a normal form game $G = \langle N, S, U \rangle$, where N is the player set, S is the strategy set, and U is the utility function.

Player Set. We consider players in the game as the processes in the committee who contribute to maintaining the blockchain ($N = \Pi$). We model colluding processes as one player p_i with voting-power $m_i \in [0, f]$.

Strategy Set. To simplify our analysis, we only consider constant strategies, i.e. strategies where players follow conduct a certain attack with constant probability every round. We analyze some additional strategies through simulation. The strategy $S(\rho, e, e_a)$ of a player is parameterized by $\rho \in [0, 1]$, $e \in [0, f]$, and $e_a \in [0, m_i]$. If a process p_i is a follower, it votes with only $m_i - e_a$ fraction of its power for a proposed block. If it is the leader, it votes with its full power. Additionally, a leader publishing a block will omit e votes with probability ρ. With probability $1 - \rho$ it will include all votes it received. Therefore, having $e > 0$ and $\rho > 0$ indicate Attack III, while Attack I is demonstrated by having $e_a > 0$.

The strategy profile $S(0, 0, 0)$ in which the players always follow the protocol is used by Altruistic processes, and is denoted by S_{honest}.

Utility Function. We define the utility function of each player as its expected payoff during a round, excluding the first T. This payoff includes both the voting reward and leader bonus. We note that due to our restriction to constant strategies, the expected payoff is constant for all rounds after the first T.

6.1 Baseline Analysis

As a baseline, we analyze the incentive mechanism introduced in Cosmos [18]. In this mechanism, the leader L_{h+1} receives an extra reward $b \times R$ as a bonus if it does include votes from all committee members. If votes from a fraction $e \leq f$ of the committee members are missing, the bonus is reduced to $b \times \frac{f-e}{f}$. We refer to this incentive scheme as the **variational bonus**. In this scheme, the expected payoff of players only depends on their behaviour in the current round. We, therefore, ignore the parameters ρ in the strategy profile and concentrate on e. In rounds, where p_i is the leader, its payoff for strategies S_{honest} and $S(1, e, 0)$ is calculated as:

$$share[\text{honest}] = m_i \cdot R + b \cdot R \quad share[\text{S(1,e,0)}] = \frac{m_i \cdot R}{1-e} + \frac{f-e}{f} \cdot b \cdot R \quad (3)$$

In order to prevent rational processes from excluding each other, the bonus must ensure that $share[\text{honest}] > share[\text{S(1,e,0)}]$. Thus, Inequality (4) must hold:

$$b > \frac{f \cdot m_i}{1-e} \quad (4)$$

Example. *If we consider $f = 1/3$, then b must be greater than $1/(2 \cdot n)$ to stop a rational process. For instance, the size of the committee in Cosmos is between 100 and 300. A block needs at least $2/3$ of the votes to be considered as approved. Therefore, according to Eq. 4, a bonus of $b = 0.005$ would be sufficient to prevent misbehaviour in individual rational nodes. The bonus of 5% employed in Cosmus is sufficient to thwart off coalitions of size up to 10%.*

Theorem 1. *If and only if Eq. 4 holds, for all m_i, the strategy profile S_{honest} is a Nash equilibrium.*

Proof. If Eq. 4 holds, $share[\text{S(1,e,0)}]$ is smaller than $share[\text{honest}]$, meaning the payoff of staying correct is more than omitting other processes for p_i with power m_i. Therefore, if all other players follow S_{honest}, player p_i cannot increase its payoff by changing e and omitting votes, when it is the leader.

Lemma 1. *The right side of Eq. 4 reaches its maximum for $e = f$ fraction of the committee.*

6.2 Collusion Resistance of Rebop

Attack III. To analyze the resistance of Rebop against Attack III, we focus on strategies deviating from S_{honest} through $\rho > 0$ and $e > 0$.

Lemma 2. *Any strategy $S(\rho, e, e_a)$ is dominated by a strategy $S' = S(\rho', f, e_a)$.*

We omit the detailed proof due to space constraints. The idea is to choose ρ', such that for $\alpha = 1$ both strategies give the same reputation. For $\alpha > 1$, S' will even give a larger reputation. Since according to Lemma 1, omitting a larger fraction is more profitable, S' gives a bigger reward.

Assume now, that all players but p_i follow S_{honest}. Further, we assume that p_i follows a strategy $S' = S(\rho', f, 0)$. If $\rho' = 0$ (i.e. $S' = S_{honest}$), the expected payoff received by p_i is:

$$payoff_i[S_{honest}] = P_{honest}[L_h = p_i] \cdot b \cdot R + m_i \cdot R \tag{5}$$

where $P_{honest}[L_h = p_i] = m_i$, since all players have reputation 1. If p_i follows S', as a payoff, it receives $\frac{m_i R}{1-f}$ reward in the rounds it is the leader and decides to omit f votes (with probability ρ).

$$payoff_i[\text{omit}] = P_{S'}[L_h = p_i]\left(\rho\frac{m_i R}{1-f} + (1-\rho)m_i R + bR\right) + P_{S'}[L_h \neq p_i]m_i R \tag{6}$$

Following S', $r_i = (1-\rho)$. This gives the following equation:

$$P_{S'}[L_h = p_i] = \frac{m_i(1-\rho)}{1 - m_i + m_i(1-\rho)} \tag{7}$$

By comparing the Eqs. 5 and 6, the bonus threshold for preventing the colluding behaviour is derived as follows:

$$b > \frac{m_i \cdot f \cdot (1-\rho)}{(1-f)(1-m_i)} \tag{8}$$

Lemma 3. *For $\rho \in [0, 1]$ the right hand side of Inequation 8 reaches its maximum when ρ is 0.*

Theorem 2. *If Inequation 8 holds, and all players but p_i follow S_{honest}, then p_i will receive a worse payoff following S' than following S_{honest}.*

Example. *Considering $f = 1/3$, in Rebop, a bonus of 0.005 is sufficient to motivate rational players not to omit votes in a system with more than 100 players with equal power. A bonus of 5% allows Rebop to thwart off coalitions of size up to 9%.*

Attack I. In Rebop, a process p_i may also try to reduce others reputation by not voting for their proposed blocks with part of its power $e_a < m_i$. We note that this attack becomes less effective if p_i itself also omits votes. We therefore analyze the payoff of strategies $S_a = S(0, 0, e_a)$. By reducing others' reputations, a process itself receives a bonus more often. However, it loses the m_a part of its reward by not voting to the approved blocks. This attack is unprofitable if the lost reward is bigger than the expected increase in bonus.

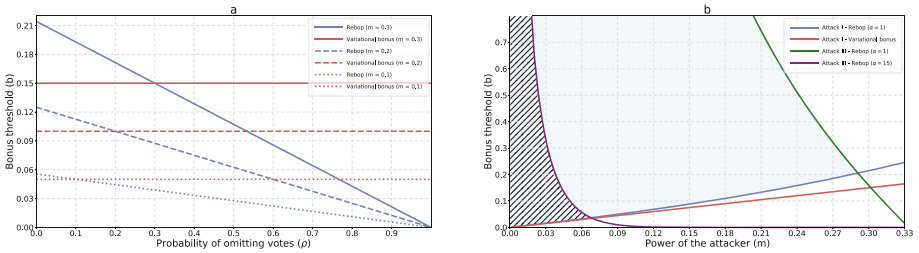

Fig. 3. a) A comparison between variational bonus (red lines) and reputation-based leader election with a fixed bonus (blue lines). The plot illustrates the minimal bonus to make omitting votes with probability ρ unprofitable for 3 different values of m_i. b) Bonus threshold for preventing Attack III for reputation-based leader election and variational bonus, and Attack I for $\alpha = [1, 15]$ and $e_a = m_i$. The blue and hatched areas show the bonuses that can tolerate both attacks together. In both plots $f = \frac{1}{3}$. (Color figure online)

$$\left(e_a - \left(\frac{m_i - e_a}{1 - e_a}\right) e_a\right) P_{S_a}[L_h = p_i] > P_{S_a}[L_h = p_i]b - P_{S_{honest}}[L_h = p_i]b \quad (9)$$

Under strategy S_a the reputation of p_i is 1, while the reputation of all correct players is $r_{c,a} = \left(\frac{f - e_a}{f}\right)^{\alpha}$. Thus

$$P_{S_a}[L_h = p_i] = \frac{m_i}{r_{c,a} \cdot (1 - m_i) + m_i} \quad (10)$$

Simplifying Inequation 9, the bonus threshold for stopping Attack I is calculated as follows:

$$b < \frac{e_a \cdot r_{c,a}}{m_i(1 - e_a)\,(1 - r_{c,a})} \quad (11)$$

The next theorem follows from the above analysis and Theorem 2.

Theorem 3. *If Inequation 11 and 8 hold, for all m_i, the strategy profile S_{honest} is a Nash equilibrium.*

Discussion. Figure 3 b) correlates the bonus size with the maximum attacker power tolerated. We see that a small bonus tolerates a similar coalition size for both analyzed methods. Nevertheless, a larger bonus is needed for Rebop to tolerate larger coalitions. Additionally, the analysis on Attack I shows that for Rebop, there exists a maximum bonus for keeping a given coalition correct. Different from the lower bound on the bonus, this upper bound depends on the value α.

Interestingly, Lemma 3 suggests that for processes with power above the threshold, the two methods differ in which attacks become profitable. This is

shown in Fig. 3 a). For Rebop, only small omissions become profitable. For example, the figure shows that given a bonus of 5% a coalition with $m_i = 0.1$ may benefit from an attack, but only if $\rho \leq 0.1$. While effective, this attack will not give a significant win. Another example is given below:

Example. *Consider $f = 1/3$, a bonus $b = 0.1$. Under Rebop, even for a coalition with $m_i = 0.33$, Attack III is only profitable with $\rho < 0.6$, meaning it is only profitable for the coalition to omit others votes 60% of time. Using Cosmos' variational bonus however, all attacks with $e > 0$ are profitable for the same coalition, meaning it is profitable to omit others in every round.*

6.3 Preventing Byzantine Attacks

None of the above schemes prevent Byzantine attackers from excluding targeted processes when they are the leader. In Cosmos, for example, Byzantine processes lose reward by attacking other processes, but it does not stop them from misbehaviour. However, different from the Cosmos' variational bonus, Rebop reduces the abilities of Byzantine attackers by prioritizing correct processes as the leaders. Assume a Byzantine process p_b with voting-power m_b in the system. Assume p_b is targeting a victim p_v with power m_v. In the schemes that use a random or round-robin leader election (Cosmos), the probability of p_b to be selected as the next round leader is always constant and proportional to its power m_b. In Rebop the probability for p_b to be the leader is reduced with its reputation r_b:

$$P[L_h = p_b] = \frac{m_b \cdot r_b}{m_b \cdot r_b + (1 - m_b)} = \frac{m_b \left(f - m_v\right)^{\alpha}}{m_b \left(f - m_v\right)^{\alpha} + f^{\alpha}(1 - m_b)} \tag{12}$$

According to Eq. 12 Byzantine attacks also on small victims (e.g. $m_v = 1\%$) can be significantly reduced by choosing a large enough α. Note that while Rebop reduces the ability of attacker to do Attack III, it gives the power to attacker for Attack I. The effect of attacks is further analyzed in Sect. 7.2.

7 Simulation Results

We conduct simulations to verify our analysis and evaluate additional situations, including Byzantine attacks. Since the committee's composition has little effect on our proposed methods, we use a constant committee in all simulations. For simplicity, we assume that all processes have an equal voting power which does not change during the experiments. We use $f = 1/3$, $|\Pi| = n = 100$, and $T = 10\,000$ in all simulations, and run for 60 000 rounds. This ensures that even with a small reputation of 0.05 a node is likely the leader at least once during T rounds.

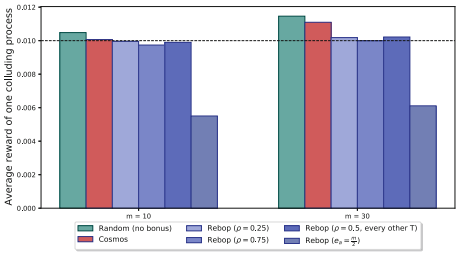

Fig. 4. Final share of one colluding process with two different coalition sizes for different attacks. In a fair environment, the share of each process is 0.01 of the total reward.

7.1 Resistance Against Colluding Processes

To show the impact of Rebop on colluding processes, we simulate Attack III by coalitions with 10% and 30% of the committee members. There is no message loss in this simulation, and the leaders receive all the votes. Bonus is set to 5% of the block reward. We evaluate Cosmos' variational bonus and the basic protocol without bonus with $e = f$ and $\rho = 1$.

We also simulated Rebop with 4 different strategies for the colluding processes: 1) $e = f$ and $\rho = 0.25$. 2) $e = f$ and $\rho = 0.75$. 3) attack every other T with $e = f$ and $\rho = 0.5$. 4) $e_a = \frac{m}{2}$.

The results of this simulation are shown in Fig. 4. It is evident that under random leader election with no bonus, even a 10% coalition can benefit from attacking the system. Variational bonus (Cosmos) makes things better, but forming a coalition in large sizes leads to a significant outcome; colluding processes can gain more than their fair share from the system by excluding any process other than themselves. However, Rebop is effective against such behavior. Even the large coalition of 30% benefits more from fewer omissions ($\rho = 0.25$). Thus, the 5% bonus is sufficient to limit attacks. In addition, because the bonus is small, Attack I is not effective. Note that attacking every other T with $\rho = 0.5$ leads to almost the same reward as attacking every round with $\rho = 0.25$.

7.2 Byzantine Resistance

We simulate the effect of Rebop in the presence of message loss and Byzantine attacks with $\alpha = 1$ and $\alpha = 15$. We also used the variational bonus (Cosmos) as our baseline. Figure 5 shows how much the resulting shares are reduced and increase through message loss and attacks. We use a bonus of $b = 5\%$. Processes exhibit different message loss and attack behavior, as summarized in Table 1. We assume that a leader with message loss loses every message with the constant probability given in the table.

Comparing shares of under attack processes, we see that Cosmos allows Byzantine processes to inflict significant harm. Consistent with our results on coalition resistance, the attackers (G7-G10) gain less than the correct G1. Rebop reduces the harm done by Attack III. G6, which has voting power 5%, loses 9%

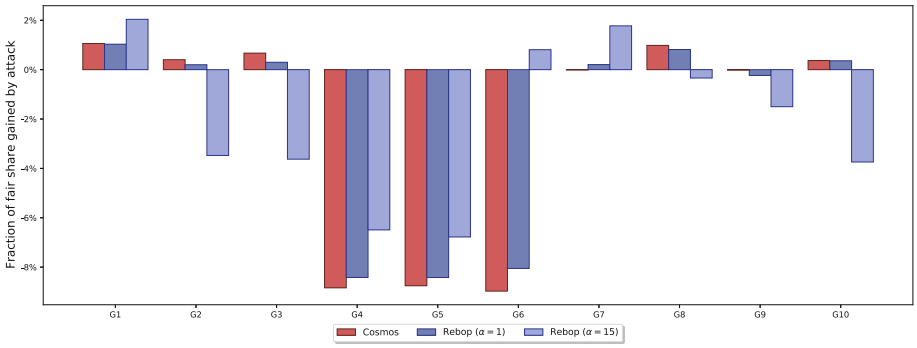

Fig. 5. The difference between final share and fair share (0.01) under message loss and attack for processes in three different configurations: random leader selection with variational bonus (Cosmos), Rebop with $\alpha = 1$, and Rebop with $\alpha = 15$. Processes are categorized into 10 groups based on Table 1.

Table 1. Summary of message loss and Byzantine attacks in the Byzantine resistance experiment.

Group	Type	Group size	Message loss	Target	Attack
G1	Correct	42	-	-	-
G2	Correct	10	5%	-	-
G3	Correct	1	-	-	-
G4	Correct	1	-	-	-
G5	Correct	1	-	-	-
G6	Correct	5	-	-	-
G7	Byzantine	10	-	G3	Attack I
G8	Byzantine	10	-	G4	Attack III
G9	Byzantine	10	-	G5	Attack I and Attack III
G10	Byzantine	10	-	G6	Attack III

of its fair share with Cosmos, 8% in Rebop with $\alpha = 1$ and gains 0.8% if α is increased to 15. For victims with smaller voting power (G5, G6), Rebop is less effective but still outperforms Cosmos. Our method still leaves some ability for attacks. That is because our model cannot distinguish between votes omitted by attackers and those omitted through message loss. We also note from G3's share that a large α opens the possibility of Attack I. To this end, the α should be carefully selected. However, even under this attack, the attacker G7 earns less than the correct G1.

8 Related Works

Fair rewarding mechanisms for blockchains have been studied for different consensus types and perspectives [11,12,30,31]. In the following, we restrict exposition to committee-based blockchains.

Lagaillardie et al. [19] studied the fairness of Tendermint in the presence of rational processes. They proposed delayed rewarding, which allows votes for a block at height h to be included and rewarded up to the height $h+\Delta$. Amoussou-Guenou et al. [2] analyzed the fairness of the rewarding mechanism used in Tendermint. They proved that the current rewarding mechanism used by Tendermint is not fair under message loss. They also proved that if a system is eventually synchronous and Byzantine behavior is detectable, an eventual fair rewarding mechanism exists for it. This differs from our assumptions, where Byzantine behavior is indistinguishable from message loss. They further extended their work in [3] to study fairness in all committee-based blockchains. They analyzed the fairness of two critical elements of committee-based blockchains: rewarding mechanisms and selection mechanisms. Liu et al. [23] proposed a fair selection mechanism for permissionless committee-based blockchains, which has two main components: the mining process and the confirmation of the new nodes list. Motepalli et al. [25] designed a framework for analyzing different reward mechanisms in PoS-based blockchains using evolutionary game theory.

All of the above works either do not consider Byzantine behavior or assume that such behavior, especially denial to receive a message, can be detected. On the other hand, FairLedger [20] proposes a detection mechanism that includes both echoing messages in case of message loss and explicit reconfiguration in case of detection.

Using reputations for different areas such as blockchain is not new. Many approaches assign a score to each user that represents the probability of that user to behave honestly [7,14,16,26]. De Oliveira et al. [29] proposed a reputation-based consensus mechanism to overcome the problem of high energy consumption. In their model, each node needs to have a higher reputation than a threshold to append a new block to the blockchain. Do et al. [10] presented an improvement for delegated PoS by replacing coin-staking with a reputation-based ranking system. Wang et al. [36] proposed a reputation-based incentive module that can be added to most consensus algorithms and help them to achieve a better consensus state. In most of the current approaches, reputations deter the reward of each process. This is different from our proposed method in which only a small part of the reward is given based on the reputation, and its main purpose is to take the ability to misbehave away from the processes.

9 Conclusion

We have analyzed different attacks on leader-based reward mechanisms in committee-based blockchains. We showed that rational processes might gain more than their fair share by building a coalition, and Byzantine processes can

reduce others' share of rewards. Then, we proposed Rebop, which uses a leader bonus and reputation-based leader election to overcome these attacks. Our analysis proves the ability of the proposed method to tackle these problems. We show that Rebop reduces the effect of Byzantine attacks, which the bonus and incentives alone do not achieve.

References

1. Aiyer, A.S., Alvisi, L., Clement, A., Dahlin, M., Martin, J.P., Porth, C.: Bar fault tolerance for cooperative services. In: Proceedings of the Twentieth ACM Symposium on Operating Systems Principles, pp. 45–58 (2005)
2. Amoussou-Guenou, Y., Del Pozzo, A., Potop-Butucaru, M., Tucci-Piergiovanni, S.: Correctness and fairness of tendermint-core blockchains. arXiv preprint arXiv:1805.08429 (2018)
3. Amoussou-Guenou, Y., del Pozzo, A., Potop-Butucaru, M., Tucci-Piergiovanni, S.: On fairness in committee-based blockchains. In: 2nd International Conference on Blockchain Economics, Security and Protocols (Tokenomics 2020) (2020)
4. Androulaki, E., et al.: Hyperledger fabric: a distributed operating system for permissioned blockchains. In: Proceedings of the Thirteenth EuroSys Conference, pp. 1–15 (2018)
5. Bano, S., et al.: Consensus in the age of blockchains. arXiv preprint arXiv:1711.03936 (2017)
6. Baudet, M., et al.: State machine replication in the libra blockchain. The Libra Association, Technical report (2019)
7. Cai, W., Jiang, W., Xie, K., Zhu, Y., Liu, Y., Shen, T.: Dynamic reputation-based consensus mechanism: real-time transactions for energy blockchain. Int. J. Distrib. Sens. Netw. **16**(3), 1550147720907335 (2020)
8. Chen, J., Micali, S.: Algorand: a secure and efficient distributed ledger. Theoret. Comput. Sci. **777**, 155–183 (2019)
9. Daian, P., Pass, R., Shi, E.: Snow White: robustly reconfigurable consensus and applications to provably secure proof of stake. In: Goldberg, I., Moore, T. (eds.) FC 2019. LNCS, vol. 11598, pp. 23–41. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32101-7_2
10. Do, T., Nguyen, T., Pham, H.: Delegated proof of reputation: a novel blockchain consensus. In: Proceedings of the 2019 International Electronics Communication Conference, pp. 90–98 (2019)
11. Fanti, G., Kogan, L., Oh, S., Ruan, K., Viswanath, P., Wang, G.: Compounding of wealth in proof-of-stake cryptocurrencies. In: Goldberg, I., Moore, T. (eds.) FC 2019. LNCS, vol. 11598, pp. 42–61. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-32101-7_3
12. Garay, J., Kiayias, A., Leonardos, N.: The bitcoin backbone protocol: analysis and applications. In: Oswald, E., Fischlin, M. (eds.) EUROCRYPT 2015. LNCS, vol. 9057, pp. 281–310. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-46803-6_10
13. Hanke, T., Movahedi, M., Williams, D.: DFINITY technology overview series, consensus system. arXiv preprint arXiv:1805.04548 (2018)
14. He, Q., Wu, D., Khosla, P.: SORI: a secure and objective reputation-based incentive scheme for ad-hoc networks. In: 2004 IEEE Wireless Communications and Networking Conference, pp. 825–830. IEEE (2004)

15. Herlihy, M., Moir, M.: Enhancing accountability and trust in distributed ledgers. arXiv preprint arXiv:1606.07490 (2016)
16. Kantarci, B., Glasser, P.M., Foschini, L.: Crowdsensing with social network-aided collaborative trust scores. In: 2015 IEEE Global Communications Conference (GLOBECOM), pp. 1–6. IEEE (2015)
17. Kwon, J.: Tendermint: consensus without mining. Draft v. 0.6, fall 1(11) (2014)
18. Kwon, J., Buchman, E.: Cosmos: a network of distributed ledgers (2016). https://cosmos.network/whitepaper
19. Lagaillardie, N., Djari, M.A., Gürcan, Ö.: A computational study on fairness of the tendermint blockchain protocol. Information **10**(12), 378 (2019)
20. Lev-Ari, K., Spiegelman, A., Keidar, I., Malkhi, D.: FairLedger: a fair blockchain protocol for financial institutions. In: 23rd International Conference on Principles of Distributed Systems (OPODIS 2019) (2020)
21. Li, W., Andreina, S., Bohli, J.-M., Karame, G.: Securing proof-of-stake blockchain protocols. In: Garcia-Alfaro, J., Navarro-Arribas, G., Hartenstein, H., Herrera-Joancomartí, J. (eds.) ESORICS/DPM/CBT -2017. LNCS, vol. 10436, pp. 297–315. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67816-0_17
22. Liu, J., Li, W., Karame, G.O., Asokan, N.: Toward fairness of cryptocurrency payments. IEEE Secur. Priv. **16**(3), 81–89 (2018)
23. Liu, Y., Liu, J., Zhang, Z., Yu, H.: A fair selection protocol for committee-based permissionless blockchains. Comput. Secur. **91**, 101718 (2020)
24. Micali, S., Rabin, M., Vadhan, S.: Verifiable random functions. In: 40th Annual Symposium on Foundations of Computer Science, pp. 120–130. IEEE (1999)
25. Motepalli, S., Jacobsen, H.A.: Reward mechanism for blockchains using evolutionary game theory. arXiv preprint arXiv:2104.05849 (2021)
26. Mousa, H., Mokhtar, S.B., Hasan, O., Younes, O., Hadhoud, M., Brunie, L.: Trust management and reputation systems in mobile participatory sensing applications: a survey. Comput. Netw. **90**, 49–73 (2015)
27. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Technical report (2008)
28. Nguyen, C.T., Hoang, D.T., Nguyen, D.N., Niyato, D., Nguyen, H.T., Dutkiewicz, E.: Proof-of-stake consensus mechanisms for future blockchain networks: fundamentals, applications and opportunities. IEEE Access **7**, 85727–85745 (2019)
29. de Oliveira, M.T., Reis, L.H., Medeiros, D.S., Carrano, R.C., Olabarriaga, S.D., Mattos, D.M.: Blockchain reputation-based consensus: a scalable and resilient mechanism for distributed mistrusting applications. Comput. Netw. **179**, 107367 (2020)
30. Pass, R., Shi, E.: FruitChains: a fair blockchain. In: Proceedings of the ACM Symposium on Principles of Distributed Computing, pp. 315–324 (2017)
31. Pass, R., Shi, E.: The sleepy model of consensus. In: Takagi, T., Peyrin, T. (eds.) ASIACRYPT 2017. LNCS, vol. 10625, pp. 380–409. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70697-9_14
32. Saleh, F.: Blockchain without waste: proof-of-stake. Rev. Financ. Stud. **34**, 1156–1190 (2018)
33. Schneider, F.B.: Implementing fault-tolerant services using the state machine approach: a tutorial. ACM Comput. Surv. (CSUR) **22**(4), 299–319 (1990)
34. Sukhwani, H., Martínez, J.M., Chang, X., Trivedi, K.S., Rindos, A.: Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric). In: 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS), pp. 253–255. IEEE (2017)

35. Vilaça, X., Denysyuk, O., Rodrigues, L.: Asynchrony and collusion in the N-party BAR transfer problem. In: Even, G., Halldórsson, M.M. (eds.) SIROCCO 2012. LNCS, vol. 7355, pp. 183–194. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-31104-8_16
36. Wang, E.K., Liang, Z., Chen, C.M., Kumari, S., Khan, M.K.: PORX: a reputation incentive scheme for blockchain consensus of IIoT. Futur. Gener. Comput. Syst. **102**, 140–151 (2020)
37. Yin, M., Malkhi, D., Reiter, M.K., Gueta, G.G., Abraham, I.: HotStuff: BFT consensus with linearity and responsiveness. In: Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing, pp. 347–356 (2019)