Lever: Breaking the Shackles of Scalable On-chain Validation

Mingming Wang*, Qianhong Wu†, *et al.*
*School of Electronic and Information Engineering, Beihang University, Beijing, China 100191
Email: wangmingming@buaa.edu.cn
†School of Cyberscience and Technology, Beihang University, Beijing, China 100191
Email: qianhong.wu@buaa.edu.cn

***Abstract*—Blockchain brings dawn to decentralized applications which coordinate correct computations without *a prior* trust. However, existing scalable on-chain frameworks are incompetent in dealing with intensive validation. On the one hand, duplicated execution pattern leads to limited throughput and unacceptable expenses. On the other hand, there lack fair and secure incentive mechanisms allocating rewards according to the actual workload of validators, thus deriving bad dilemmas among rational participants and inducing effective attacks from shrewd adversaries. While most solutions rely on off-chain patterns to sidestep the shackles, it further introduces unexpected issues in applicability, fairness and brittle dependency on interactive cooperation. The intrinsic bottleneck of backbone has never been drastically broken.**

This work presents Lever, the first scalable on-chain framework which supports intensive validation, meanwhile achieves validity, incentive compatibility and cost-efficiency tolerance of $f < n/4$ Byzantine participants. Lever firstly integrates the evaluation of complexity into the correctness of transaction, thoroughly decoupling intensive validation from regular Byzantine consensus. Significant scalability is then achieved by launching few rounds of novel validation-challenge game between potential adversaries and rational stakeholders; compelling incentive mechanism effectively transfers deposits of adversary to specialized rewards for honest validators, therefore allows the user to lever sufficient endorsement for verification with minimum cost. Combined with game-theoretic insights, a backstop protocol is designed to ensure finality and validity of the framework, breaking through the famous Verifier's Dilemma. Finally, we streamline Lever under the efficient architecture of sharding, which jointly shows robust to conceivable attacks on validation and performs outstanding ability to purify Byzantine participants. Experimental results show that Lever vastly improves the throughput and reduces expenses of intensive validation with slight compromise in latency.

I. Introduction

In modern computation and application systems, it is difficult to obtain reliable public services without a trusted third party. Public blockchain thrives to remove such dependency with the help of permissionless consensus and ingenious incentive mechanism. Furthermore, the innovation of smart contract endows it with the potential to subvert most existing application architectures, processing complicated transactions in a fair and ordered manner.

However, the usability is far from the expected in practice, where thousands of nodes consume tremendous power to bear the functionality no more than a laptop. For fear of decentralization and restricted by duplicated execution pattern, the complexity of transaction is confined to an interior scope. The performance of backbone scarcely increases with the capacity of node, which unfortunately matches up the Liebig's law[1]. Although advanced on-chain frameworks based on Byzantine Agreement [1]–[4] plus Sharding [5]–[10] achieve sub-linear scalability, they have to assume negligible workload of validation in consensus, get vulnerable and inefficient when dealing with complicated operations. Admittedly, incompetency in supporting scalable intensive validation has become Achilles' Heel of current blockchain systems.

Advanced research trend concentrates on off-chain solutions to circumvent the problem. By moving data and validation elsewhere off the blockchain, effective schemes like the signature oracle [11], [12] and challenge-response game [11], [13], [14] are proposed to mitigate the pressure of backbone. These works made significant optimizations, albeit a series of compromises are introduced:

- Most schemes involve frequent interactive operations in state transmission or dispute resolution, which considerably entails cooperative behaviors of participants (In many cases [11], [12], [15], unanimity is required). This yields to poor robustness, where adversary could consume little cost to halt the procedure and honest nodes are vulnerable to offline attacks. Thus, the finality of validation can hardly be guaranteed.
- There may lack compatible incentive for rational validators [11]–[14], [16], [17], which could cause unreliable motivation in protocol execution. For instance, once a dispute is triggered, excessive expenses and trivial revenue could easily make the disputer give up, violating the correctness of verification.
- Since Sybil-Attack resistant mechanisms are difficult to be deployed off-chain, threshold measures may fail to take effect. Some proposal [16] suggests to directly extract sub-

[1] Just like a chain is only as strong as its weakest link, the capacity of current blockchain ecosystem seems to depend on the bottom line of a single node.

stantial incumbent on-chain validators. However, this may disturb the order of the backbone consensus.

- Some solutions [11], [12], [15] require a prior registration and a moderate member size which could limit their universality to deploy most public applications with dynamic membership.
- Strong assumptions like trusted third party and honest participants [15] may also be imported.
- Still, all schemes resort back to the backbone for a number of functionalities like state anchoring, deposit management and probably dispute resolution, which suggests the inefficiency of on-chain validation would always be the inherent bottleneck of its auxiliary solutions.

A. *Overview of our protocol*

As the above intractable restrictions hinder the off-chain techniques from thoroughly tackling the problem, we focus back to the design of backbone, aiming to propose a scalable validation framework supporting intensive workload, meanwhile achieves the desired properties of validity, finality, incentive compatibility, cost efficiency and perfect applicability. Therefore, we gradually make the following contributions.

First, we define the validation and its verifiers in a refined and practical manner. Inspired by Chainspace [10], the asymmetric execution model is utilized to simplify validation from burdensome computation, where user derives the state of contract off-chain and embeds the necessary data[2] to the transaction. Validation is then reduced to a deterministic decision problem, which also reduces the diversity of malicious propositions. While complexity is viewed as the vital element to carry out secure and fair computation [20], [21], we define a *Validation Intensive Transaction* (VIT) as a transaction of which validation overhead is beyond a fixed complexity bound.

In Lever, a validator is obligated to predicate the validity of certain transactions. Considering his behavior could be possibly determined by the corresponding reward and computation overhead while handling non-trivial workload, we adopt the Byzantine-Altruist-Rational (BAR) model [22] to describe such phenomenon. Therein, nodes are rational when validating a VIT, taking maximization of payoff as their priority, and are assumed to be honest while undertaking ordinary workload from consensus (e.g. ledger transfers). There also exists $f < n/4$ Byzantine participants in system, who could behave arbitrarily to break the protocol.

Second, we propose the Lever-Boost game, a novel validation-challenge pattern specialized for VIT processing, in substitution for the duplicated execution pattern deployed in most blockchain systems. In most existing public consensus, every potential member is required to independently verify every proposed transaction, which greatly limits the efficiency of consensus and imposes unacceptable expenses on user. To address the issue, an intuitive way is to curtail the scale of validation, in which the ideal case is to facilitate secure single verification with certain disinterested validator(s). The classical challenge-response protocol modified by Arbitrum [11] and Truebit [13] makes impressive effort to achieve the goal. Unfortunately, it is troubled with clunky interactions, lengthy latency, rigorous online requirements and a challenge interface prone to abuse, which make it fail to employ on-chain. By contrast, our pattern removes the above dependencies:

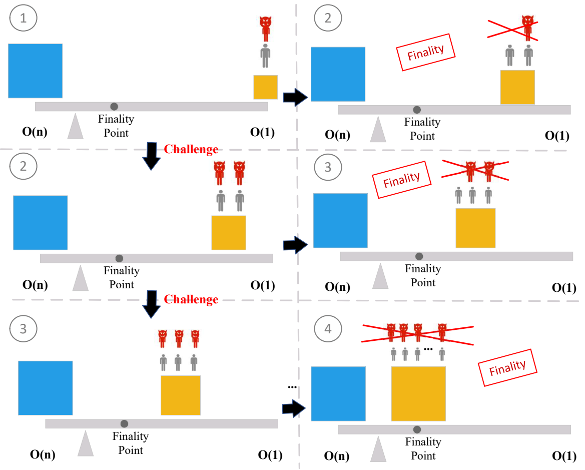


Fig. 1: Intuitive Idea of Lever

Let the initial reward, deposit and the execution time-bound of a VIT stay linear with its complexity. Illustrated by Figure 1, a transaction founder only needs to provide the reward enough for a single validation to bootstrap the game. Then, a random incumbent validator is assigned to accomplish the task, lock sufficient deposits to endorse for his verdict within the time bound. As the endorsement is anchored, any node including the stakeholders of VIT could propose their challenges to endorse for the opposite verdict, reviving the game of next round. Every new round doubles the deposit required by endorsement, meanwhile leads to an exponential increase of incentive attached to the VIT[3].

Finality could be achieved in two ways. Once there exists no adversary or the adversary's budget exhausts, no challenge would be proposed within the time bound. The game will be efficiently terminated with the correct verdict. Otherwise, when there is a stubborn adversary consistently delaying the procedure with bribery and false challenges, the game quickly converges to meet the incentive demand of a duplicated validation. Elaborate backstop scheme could promote agreement on the validity of transaction in a Sybil-resistant committee. This not only guarantees the validity of game, but also ensures the validation be always finalized within a finite latency.

In either case, the forfeits of malicious nodes satisfy the specialized incentive for every rational verifier and challenger, attracting increasing rational disinterests to faithfully carry out the validation. Even if honest stakeholders exhaust their stakes in the game, validators help to challenge wrong verdict due to

[2] In general, complete inputs and outputs are required to initiate the deterministic validation program. When privacy is crucial, schemes like zkSNARKs [18], [19] could also be used to generate secure cryptographic proofs.

[3] Because of someone between the challenger and verifier has made the wrong verdict, his deposit would be forfeit to upgrade the incentive of transaction.

their rationality w.h.p. Thereby, any censorship attempt could be unrealistic and ineffective, even supported by an extremely wealthy adversary. As for the stakeholders, their demands on deposit, online and interactivity are significantly relaxed.

As the Lever-Boost game entails a deterministic incentive, we propose the hardness model, integrating the accurate measurement of complexity (defined as hardness η) into the validity of transactions. Using hardness as the benchmark, rigorous control is further exerted on incentive and the execution time bound. After off-chain computation, transaction founder is urged to faithfully record η and accordingly associate enough incentive while creating the transaction. Hence, nodes can always get fair incentive pertaining to their actual workload. Meanwhile, deposit could be used as a powerful metric in quantifying the capacity of a validator. Combined with the incentive mechanism above, we can prove the incentive compatibility of our scheme.

Third, we propose the dual channel model to make Lever-Boost game compatible with the existing backbone frameworks. In Daily channel, classical Byzantine agreement is operated to anchor the state of VIT and its attached endorsements, while the cumulative view number is used to keep track of the execution timeout. In Lever channel, selected validators efficiently handle every intensive workload decoupled from the Byzantine consensus, making the whole construction scalable and safe.

Fourth, we design a backstop protocol to resolve the possible disputes in Lever-Boost game, achieving secure intensive validation under Byzantine-Rational (BR) assumption following the duplicated pattern. In Lever channel, a specialized voting game is organized to trace the verdict of each member. Commitment scheme is utilized to thwart the attempts of freeloading, and an appropriate threshold is established to ensure the validity of protocol. Inspired by game-theoretic insights [23]–[28] and the employment of restricted punishments [29], the protocol is adjusted to facilitate honest validation as the unique dominant strategy, figuring out the famous Verifier's Dilemma [30].

Lastly, we instantiate the above designs under the architecture of sharding [7], [8], [10], which provides linear optimizations on communication, storage and execution over ordinary workload. Correspondingly, we adapt our construction to be atomic, independently deployable in every shard. Efficient modules for incentive management, task allocation and state anchoring are created to streamline the framework. Self-enforcing reconfiguration is employed to purify Byzantine participants. Also, a flexible interface is designed to facilitate the off-chain schemes with better fairness and applicability.

To summarise, our contributions are as follows:

- We propose the hardness model and dual channel model, which combine to drastically decouple the intensive validation from classical Byzantine consensus, additionally provides deterministic guarantees for incentive and time-sensitive events.
- We create the Lever-Boost game, the first scalable backbone validation pattern which supports intensive workload, meanwhile achieves incentive-compatibility, cost-efficiency and perfect applicability.
- To ensure the validity and finality of the game, we propose a backstop protocol following the duplicated validation pattern, which provably solves the Verifier's Dilemma.
- We deploy our framework under the architecture of sharding, streamlining the design by presenting efficient algorithms of atomic execution, hierarchical election, incentive management and workload coordination.
- We provide convincing proofs and build a proof-of-concept implementation to evaluate our instantiation. Experimental results suggest that, in dealing with intensive validation, Lever linearly scales the throughput of system with the increase of nodes, and resolves over 96.5% workload via single validation pattern even in the worst-case configuration. Let n denote the number of incumbent validators in backbone protocol, it reduces expenses of user by at least $\log n$ times with slight latency imported. In term of ecological fitness, Lever provides sufficient incentive to rational validators and efficiently eliminates internal Byzantine participants while staying adaptive to the budgets of external adversaries. We compare Lever with state-of-art solutions in Table I, which shows the comprehensive advantages of our framework.

II. Background and Related Work

In this section, we firstly review the famous Verifier's Dilemma and its extensions over various kinds of backbone consensus. Then, we present a comprehensive survey over existing solutions and their limitations. Finally, we introduce the consensus scheme of Solidus [2], [31] and the game-theoretic tools which our framework builds on.

A. Instrinsic Backbone Shackles

1) Classical Mining Trouble: Practical blockchain ecologies like Bitcoin [32] and Ethereum [33] keep to an ideally duplicated pattern for validation, which requires every full node in the network conduct accurate deduction on the current block state. The enabled consensus takes the computation-intensive mining as the exclusive standard of reward distribution, where the winner takes all while others obtain nothing. Such a pattern works well if the overhead of consensus stays negligible. However, when the evolution of smart contract proposes more workload on validation, there appear catastrophic security and performance issues.

Luu et al. [5] initially point out the inharmonious phenomenon and name it the Verifier's Dilemma. Briefly speaking, by acting altruistically, exploding cost on validation would only make nodes distract and lose their mining reward. While by acting rationally and skip heavy verification, the safety and consistency of scheme would be undermined by invalid transactions and frequent forks. To avoid the dilemma, Ethereum Community restricts the complexity of transactions by setting GasLimit, a miner-defined workload upper-bound, which totally sacrifices the liveness of VIT and the usability of smart contract. But still for this, the backbone of Ethereum is

TABLE I: Comparison of Lever with state-of-the-art solutions*

Protocol	Ecology	IC[1]	Complexity[2]	Assumption	Finality[3]	Cost	Interactivity	Applicability
Exact Com [30]	Open	No	$O(n)$	BAR	$O\left(\frac{P_g}{P_l}\right)$	nX	N	Restricted
Appro Com [30]	Open	No	$O(n)$	BAR	$O(1)$	nX	N	Restricted
State Channel [12]	Closed	No	$O(1)\mid O(n)$	BA	$O(1)\mid O\left(\frac{P_g}{P_l}\right)$	$0\mid nX$	Y	Restricted
Arbitrum [11]	Closed	No	$O(1)$	BA	$O(1)\mid Inf$	$0\mid \frac{P_l}{P_g}X + \log(\frac{P_g}{P_l})F$	Y	Restricted
Truebit [13]	Open	No	$O(1)$	BAR	$O(1)\mid Inf$	$X\mid X + \log(\frac{P_g}{P_l})F$	Y	Complete
Yoda [16]	Open	No	$O(k')$	Semi-honest	$O(1)$	$k'X$	N	Complete
BDR [15]	Closed	Yes	$O(1)$	BAR+TTP	$O(1)$	$2X$	Y	Restricted
Lever	Open	Yes	$O(1)\mid O(\log n)$	BAR	$O(1)\mid O(\log\log n)$	X	N	Complete

* Let P_g denote the total instructions to execute the verification program, the constant P_l denotes the amount of instructions that current backbone could mostly afford. As for payoff, X denotes the expense (including execution cost and reward) to carry out a single verification, F denotes the transaction fee per time to anchor a digest to the backbone. We use $A|B$ where A denotes the average case, and B refers to the worst case. In Yoda's scheme, k' denotes the number of nodes in an execution set.

[1] Incentive Compatibility.

[2] The times of validation required to finalize a VIT in each framework.

[3] The rounds expected to achieve finality, this also reflects the latency of each framework.

beset by restricted capacity, censorship [34], goose egg [35], [36] and expensive transaction fees.

By smartly decoupling validation from leader election, ingenious frameworks [34], [37]–[39] employing PoX consensus [40] partially eliminate the conflict, some [37], [38] also build explicit incentive schemes for verifiers. Unfortunately, these protocols could not remarkably raise the capacity of validation due to inherently poor scalability of PoX. Additionally, it is hard to completely avoid the negative impacts on verifier's utility caused by forking.

2) Challenges from Strong Consistency: To overcome such problems, classical Byzantine agreement [41]–[43] is introduced as the building block of backbone consensus. Without loss of generality, frameworks deploying committee-based consensus [1]–[4], [31], [44] primarily use powerful anti-Sybil-Attack algorithms to determine the identities of committee members, then efficient Byzantine agreement are used to process the validation workload in permissioned environment. Strong consistency is attained to curtail the confirm latency of transactions within single consensus round. Additionally, the breakthrough of sharding-based consensus [5]–[8], [10], [45] achieves linearly scalability on throughout w.r.t. the number of participants. Through devising the blockchain state into several independent shards, the protocols allow multiple committees to run on parallel, efficiently handle the workload. An atomic inter-shard protocol is required to maintain the consistency and safety of cross-shard transactions. Remarkably, Chainspace [10] proposes elegant sharding designs to support smart contract.

Although the advance appears heart-stirring in relieving the Verifier's Dilemma, the activation of strong consistency actually draws more intractable issues into the context. Only few researches [16], [22], [46] have made fragmentary discussions on this point. Thus, we present an in-depth summary from the following two aspects:

Lethal Nondeterministic Elements. In nature of practical BFT state machine replication [41], [43], validity is left outside the protocol, making it fail to deterministically assert the validity of a transaction from the failure of consensus. Intensive workload amplifies the harm of such effect. Adversaries and selfish nodes could easily exploit non-determinism to sabotage the efforts of honest verifiers. For instance, a DoS attack which broadcasts massive VITs could make the leader's valid proposition suffer timeout expires during the Prepare phase, finally rise up to an inequitable view change. Censorship aiming at some conflicting VITs would easily make honest nodes' validation worthless. In addition, lazy nodes could always freeload the altruists' verdicts without any risk. Existing researches [22], [47]–[49] have presented elaborated measures to detect and remit the above Byzantine Faults. Unfortunately, the solutions are either too costly to deploy [47]–[49] or even impractical under the permissionless nature of public blockchain [22].

Brittle Altruist Dependency. Another severe issue falls on the Byzantine-Altruist (BA) assumption used by BFT protocols. In that, it neglects the pressing need of incentive to maintain the vitality of public blockchain. Whereas the Byzantine-Altruist-Rational(BAR) assumption [22] proposed by Aiyer et al. serves a more practical choice, especially when the operation costs should not be ignored. However, nearly all of on-chain frameworks sidestep the design of incentive mechanism and treat verifiers as altruists, only Solidus [31] make positive attempts by setting message-reply rank in consensus as the standard of reward distribution. Unfortunately, as the rank itself is a lethal nondeterministic element[4], the mechanism is still vulnerable and unprovable. Manshaei et al. [46] commit game-theoretic analyses[5] on the existing sharding-based systems. In their evaluation, if the reward is uniformly distributed, rational nodes could defect by just idling to freeload the rewards. The game will then evolve into a variant of Social Dilemma [50], [51] with the unique equilibrium that everyone defects. The replica will tragically lose its liveness and stuck in endless view change.

[4] The reported ranks can be forged without leaving any accountable evidence.

[5] The analyses are also adaptive to the committee-based consensus with trivial adjustments.

Unlike [46] that only considers the abort case, which derives a lose-lose situation to everyone in the committee, we propose a more tricky and aggressive strategy for potential rational players to always get the best payoff by plagiarizing other's responses[6]. While in this time, the replica regains its liveness but potentially loses its validity when adversaries become rational [52] or launch attacks from collusion. Since the strategy is undetectable and extremely effective in practical, we elaborate its mechanism and carry out detailed analyses in Appendix A.

In consequence, existing scalable on-chain frameworks are even more fragile confronting intensive workload. To support scalable validation of VITs in the BAR model, underlying challenges should be overcome: 1) Decouple intensive workload of VIT from Byzantine consensus. 2) Eliminate nondeterministic elements with every effort. 3) Set rigorous reference for incentives and bring game-theoretic insights into the framework.

B. Limitation of Existing Solutions

While the Verifier's Dilemma remains intricate, rare efforts have been made to the on-chain verification. Only Luu [30] presents two potential approaches: the exact consensus computation to split the overhead of validation into multiple sub-transactions, the approximate consensus computation to commit probabilistic validation through the idea of sampling inspection. Unfortunately, both mechanisms are only applicable in applications with perfect decomposability. Excessive delay and risks on biased randomness further breaks their feasibility.

Whereas diverse patterns [53] significantly reduce the workload by resolving the overhead of computation and storage off-chain, only necessary data need to be uploaded for proving the correctness of execution. Impressively, powerful schemes like zkSNARKs [18], [19] could simplify arbitrary computations to a succinct proof with bounded complexity, meanwhile strengthens the privacy in validation. However, massive off-chain overhead is additionally introduced for proof generation, which is definitely not cost-efficient. State-of-the-art solutions are proposed to further simplify or carry out the validation off-chain, according to the following mechanisms:

1) Oracle among Stakeholders: With predefined membership in this pattern, an assertion oracle can be built by collecting valid signatures from stakeholders of contract. Then, the transition of states is automatically pushed by unanimous assertions, despite the validity of proposition. Better performance is achieved owning to the limited membership, and cooperation becomes the paramount element rather than incentives. The elegant framework, General State Channel [12] follows exactly this pattern. Unfortunately, if there appears any disoperative behavior, all uncompleted transitions have to be executed back on chain, incurring the worst efficiency and unacceptable expenses.

To resolve possible disputes off-chain, Arbitrum [11] adopts a challenge-response bisection protocol above the pattern. When dispute happens, each stakeholder could challenge the current proposer within a pre-defined timeout. If it expires, the proposed transition is viewed as valid. Otherwise, the proposer should bisect the execution and submit the hash of two halt states in each subspace, while the challenger should choose the earliest discrepant one to challenge again. For an execution involved N steps, each cast of challenge repeats for at most $\log(N)$ rounds and convergences to a single step of execution, which is toleranted to verify on-chain. Note that, massive interactive states of the game should be anchored on-chain, which incurs considerable latency and expenses. Though Arbitrum proposes to carry out the bisection protocol on state channel to optimize the performance, such construction is inherently vulnerable to uncooperative behaviors, trapped by the chicken-and-egg dilemma. Things become worse when defectors could infinitely abuse the challenge interface to delay the proposition until their budgets run out. The dense interactive operations also bring members great trouble while being off-line [54], an adversary can exploit this to win the game.

The applicability of this pattern remains pretty restricted as robustness and efficiency deteriorate sharply with the increment of stakeholders. The pattern also cannot handle applications with varying membership, public competition, or potential procedures incurring violent conflict of interest.

2) Construction of Open Ecology: This pattern manages to build a self-governed open ecology off-chain by importing disinterested validators and dependable incentives. Truebit [14] and its groundwork [13] combine the aforementioned challenge response bisection protocol with abundant incentive measures such as community driven jackpot, verification tax, a reward distribution scheme countering excessive competition. Unfortunately, it is vulnerable to the Sybil Attack, causing incompatible incentives, which is catastrophic in open ecologies[7] [11]. Briefly speaking, an adversary could generate several identities and burst to validate one task, making the incentive become negligible to rational verifiers. Thus, a provably bad equilibrium derives making the task lose its validity.

Yoda [16], as another solid solution, circumvents the Sybil Attack by directly extracting validators from backbone and assigning tasks randomly with a compulsive mode. Validation is executed sequentially among several groups of disinterests, while standards from Hyptothesis Testing are used to control the termination of each task. Effective schemes are proposed to restrain the freeloading within or among groups. However, by releasing the BAR model assumption to Quasi-Honest, the protocol avoids discussing incentive in detail. This boils down to incompatible incentives if termination cannot be attained as expected within certain groups. Besides, harshly-managed forfeits could weaken the enthusiasm of verifiers for participation. Also, incomplete workload isolation may incur negative effect on backbone consensus.

[6] Thus, we name the derived phenomenon, the Faineant's Revelry.

[7] The issue is formally discussed in Arbitrum [11], namely the Participation Dilemma.

3) Game Theoretic Solutions: Resorting to game theory, some patterns manage to facilitate cooperative behaviors and achieve incentive compatibility off-chain, while backbone is viewed as a dependable abstraction to conduct incentive management. However, there are obvious restrictions in these patterns. Dong et al. [15] designs elegant sequential games to restrain two rational solvers from colluding and cutting corners, additionally achieving validity and cost-efficiency in intensive outsourced computing. Unfortunately, it assumes the existence of a trusted third party, also the task giver must be honest. SmartCast [17] builds an off-chain ecology supporting the execution of smart contract under the BAR model, but it neglects the cost of validation, let alone many plausible attacks ruled-out in the protocol.

Through discussions over various off-chain patterns, more valuable desires are left to our research: 1) Build an anti-abuse challenge protocol with no dense interactions . 2) Facilitate fair and incentive-compatible competition under open ecology. 3) Bridge the cost-efficient single validation pattern to robust duplicated pattern among disinterests while keeping respective merits.

C. Preliminaries

1) Solida: Solida [2], proposed by Abraham et al, is a committee-based consensus protocol which eliminates the negative effects of fork and selfish mining on committee reconfiguration. Other than the use of Nakamoto consensus in common constructions [1], [3], [4], committee election is done by a Byzantine consensus protocol lead by external miners. As a result, reconfiguration can be achieved with a fast and deterministic confirmation, deriving a fresh and safe membership for the underlying committee. Meanwhile, internal members consistently lead regular consensus on batches of transactions following the round robin manner.

To enforce a total order among both kinds of leaders, Solida resolves the *view* of consensus with a configuration-lifespan-view tuple (t, e, v). Each new round of usual consensus makes a member increases its view number v by 1. On receiving a new PoW for current configuration, every member increases the lifespan number e by 1, and resets v to 0. When a reconfiguration event is finally committed, every member increases the configuration number t by 1, and resets e, v to 0. Leaders are ranked by t first, then e, and then v. And a deterministic function $\mathsf{pk} \leftarrow L(t, e, v, \{\mathsf{pk}\})$ is set up to derive the unique leader of *view* according to the current committee configuration $\{\mathsf{pk}\}$.

Concretely, the Byzantine consensus in Solida is similar to the standard construction of PBFT [41]. After the agreement is reached, a special stage of *Notify* is introduced to broadcast the consensus decision to the whole P2P network. External miners can therefore retrieve the latest state of blockchain. As for the election consensus, the miner is firstly required to broadcast his PoW certificate to the committee. Then, he synchronizes and re-proposes the current state of consensus by collecting valid $2f + 1$ status messages from incumbent members. If the preposition is finally committed, he would substitute as a new committee member. Also, a blockchain checkpoint h and a fresh random seed rnd are derived for the next election. Formal proofs and details can be obtained in [2].

Solida is used as the building block of Lever since it not only performs perfect liquidity and efficiency on committee reconfiguration, but also offers a stable, frequent and relatively cheap random source. This provides the crucial guarantees of fairness in our design. We abstract the two types of consensus in Solida, the function of Daily Byzantine Consensus is simplified as: $\mathsf{state} \leftarrow \textbf{DailyBFT}(\{\mathsf{tx}\}, \mathsf{view})$. And the function of Daily Election Consensus is in the form of: $\mathsf{rnd}, \{\mathsf{pk}\}, \mathsf{h} \leftarrow \textbf{DailyEL}(\mathsf{PoW}, \mathsf{view}, \mathsf{state})$.

2) Insights from Game Theory: Under the adversarial environment of public blockchain, intensive validation brings new challenges on the incentive design and analyses of Lever. Therefore, we employ the following game-theoretic concepts and tools to make the corresponding refinements:

First, we fit the scenario of Lever as a non-cooperative game, where there exists no external authority enforcing cooperative behavior. Rational nodes distrust each other, compete independently and can hardly group into coalitions to share truthful information. In a validation game, we aim to propose a dominant-strategy-incentive-compatible (*DSIC*) mechanism [26] that honest validation stays as a weakly dominant strategy for every rational player. This means regardless of what any other players do, the strategy earns a player a payoff at least as high as any other strategy.

It is noticeable that a lazy but rational player could totally save the cost of intensive validation by guessing the validity of a transaction. Since classical Nash equilibrium does not take computational concerns into account, we employ the Bayesian Machine Game [24] proposed by Pass and Halpern to cope with the influence of such randomized strategy. The framework constructs Turing machines for strategies to include the cost of computation. A Bayesian game is then formed to return a distribution over actions. Expected utility of certain strategy can thus be derived from the result.

In addition, an equilibrium is defined as t-immune [23] if the non-deviating players are not made worse off by arbitrary (possibly coordinated) deviations by up to t players, which captures the protean nature of Byzantine behaviors in our game.

Lastly, when there exist two or more equilibriums at the same time, Harsanyi's theories [55] are adopted to estimate the theoretical probability of any given equilibrium to emerge as the outcome of game. This theory has been refined and applied to many similar circumstances [27], [28], [56]–[58] and the computation of particular mixed-strategy equilibrium always acts as the key approach. With reference to the studies of restricted punishment [29], [59], negative equilibriums could be further eliminated by tracking down the deterministic evidence of deviating strategies.

III. Model and Problem Definition

A. Network Model

Similar to prior works [1], [5], [8], [32], we assume a well-connected network with dynamic membership in partially synchronous model [41], where messages are due to arrive within a prior loose upper bound Δ and an unknown actual delay δ approached by the optimistic, exponentially increasing time-outs. Computational puzzles [32], [60] are utilized to generate Sybil-resistant identities for validators at a stable rate. Nodes consider the public/private key-pair as their unique identities and set up secure authenticated channels between each other. Also, they are assumed to have equivalent computation resources. Finally, we assume the existence of random oracle.

B. Threat Model

We adopt the BAR threat model and assume there exist n incumbent validators in our system, which contains at most $f < n/4$ Byzantine nodes. These malicious nodes can behave arbitrary to deviate from the protocol, including but not limited to abort, freeloading, taking bribes or collusion. The adversary is computationally bounded and have limited budget. In addition, it is slowly adaptive [1], [4], [7], which indicates that no altruists or rational nodes would be corrupted between reconfiguration events. We further assume that at least $3n/4$ non-Byzantine validators stay online under any circumstance. Inspired by the construction of ϵ-consensus computer [30], the rationality of nodes are defined by the complexity of validation: When dealing with transactions beneath a fixed complexity bound η', nodes are assumed to be altruistic. Classical Byzantine consensus are thus secure as the groundwork of our design. Otherwise, the nodes are rational and would take foremost interest as maximizing their payoffs.

In an intensive validation, the total utility $\mathcal{U}$ of each party is assumed to depend on the following equation: $\mathcal{U} = \mathcal{I} - C + \mathcal{B}$. Therein, $\mathcal{I}$ denotes the incentive he obtains from Lever (possibly be a reward or a forfeit), C denotes the computation cost he consumes on the intensive workload and $\mathcal{B}$[8] refers to the extra benefits outside the protocol once the certain verdict is finalized (e.g. bribery received by the Byzantine or potential business revenue brought by the VIT itself).

On this basis, we assume w.r.t. each VIT, except for the transaction founder, there exists at least one rational stakeholder who has the ability to complete the validation[9]. Our design is simplified from external economic risks by assuming the relatively stable currency and the presence of sufficiently full-incentive transactions [61].

C. Problem Definition

We assume a set of VITs are sent to our protocol. Let $c_i.v()$ denote the deterministic checker with respect to the transaction Tx_i, S_h act as the strategy to honestly conduct the validation and λ be the security parameter. Lever outputs the valid VIT set $X = \{Tx_i\}$, where $|X| = l$, and supports the following properties:

- *Incentive Compatibility:* Lever ensures the intensive validation is *DSIC* [26] on-chain. In other words, all validators who follow S_h and broadcast their faithful results will get the best and deterministic positive payoffs, on the contrary, any strategy that deviates from the protocol will end up with a negative payoff.
- *Validity:* For every transaction in the output set X, $\forall i \in \{1..l\}$, $c_i.v(Tx_i) = true$.
- *Finality:* Each VIT will get finalized within δ_f with a high probability of at least $1 - 2^{-\lambda}$.
- *Scalability:* Throughput of system on intensive validation, grows linearly with the increasing of n.
- *Agreement:* All non-Byzantine nodes in the system agrees on the validity of VITs within δ_s after its finalization.
- *Cost-efficiency:* The founder of VIT only needs to provide one piece of reward to proxy the workload on-chain, which is close to the actual execution cost of validation.
- *Self-reinforcement:* Lever adaptively provide lower latency and redundancy on validation with the decrease of adversary's budgets. Furthermore, effective purification is autonomously carried out on incumbent Byzantine nodes.
- *Efficient:* Each VIT only incurs a single time of verification on average and $O(\log n)$ times to achieve finality in the worst case, the overhead of storage and communication can be optimized by the full-fledged architecture of sharding.

IV. Main Protocol

A. Decouple of Intensive Workload

1) Dual Channel Model: Bordered by the fixed complexity bound η', we build two channels keeping to different validation patterns, of which skeleton interactions could be viewed in Figure 2:

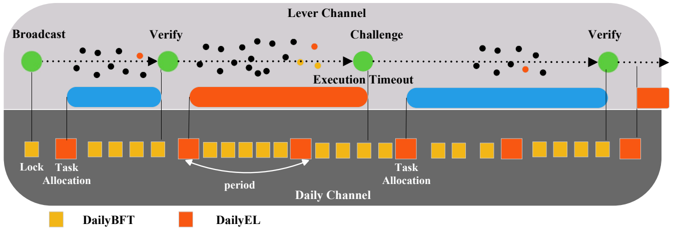


Fig. 2: Dual Channel Model

In *Daily Channel*, the Byzantine consensus[10] of Solida is deployed to handle transactions and events with negligible

[8] In some sense, the parameter accounts for the rationality of stakeholders to dispute and adversaries to deviate, making our model more realistic and universal.

[9] The assumption can be released by seeking for a rational third party to proxy the decision of challenge. Such escrow-driven design can be inspired by Pisa [54] and BDR [15], our scheme leaves attractive incentive for disinterested challengers, which is conducive to future improvements.

[10] Other committee-based schemes like Hybrid Consensus [53], ByzCoin [39], Algorand[30] etc. can also be appropriate for Lever with slightly change on workflow. However, extra latency will be induced during wedging and task allocation.

overhead under the BA assumption. Hence the execution is fast and safe with the duplicated pattern. An oracle is set up to conduct efficient management on identities and incentive of validators. Specifically, *DailyBFT* provides a fair and immutable environment to timely anchor typical audit states from intensive validation. Whereas *DailyEL* sets up a global clock by its cumulative configuration number t, which deterministically measures the execution timeout of a VIT. Meanwhile, the newly derived rnd is employed to achieve unbiased task allocation. We denote the interval of election consensus as a *period* δ_p.

In *Lever Channel*, disinterest nodes make their choices out of rationality. An open and competitive ecology is established to decompose the intensive validation into finite rounds of verification-challenge game. In each round, validation will be taken only once, the winner's proposal will get confirmed fairly and rapidly by Daily Channel.

Concretely, the Verify proposal is only raised by one specified incumbent validator, ensuring best performance and orderliness, while the Challenge interface is open to the whole ecology, minimizing the effects of censorship. An execution timeout δ_e, determined by the complexity of VIT, is initiated to control the time limit of both proposals. Also, a specified deposit is attached to each proposal, gathering more incentive for VIT with the increment of rounds, as well as reducing the abuse of challenge interface. Rational disinterests and stakeholders protect against malicious proposals by launching challenges. Once the adversary fails to resist, the VIT will be finalized efficiently. When the incentive is sufficient to afford m nodes to undertake the validation, the backstop protocol will be activated under the current configuration of committee, taking over and resolving the dispute within $\delta_e + \delta_p$ under the BR assumption.

2) Hardness Model: In our design, it is catastrophic when there exist non-deterministic elements on the complexity and incentive of transactions. On the one hand, adversary could flood excessive VITs with little reward [13] to disincentive faithful validation from rational nodes, or he could create goose eggs [35], [36] to sabotage the fairness of game. On the other hand, misstatement of complexity, which is perfectly possible in single validation, could totally make the Dual Channel Model trivial. It further leads severe secure and incentive vulnerabilities.

To address such issues, we firstly define the hardness of a transaction, η, as the precise measurement of its validation complexity weighted by a predetermined instruction set. Other than the gas system of Ethereum [33], [62][11], the accuracy of hardness is integrated into the validity of transaction. In this way, misstatements on η only result in the invalid transactions to user and wrong proposals to potential validators or challengers. By comparison, our design creates an additional overhead off-chain since user is required to measure and record η before broadcasting transactions. It could be negligible when determinate transition could be achieved by computation [14], otherwise it is equal to the cost of validation.

Using hardness as the benchmark, we further exert rigorous control on reward, deposit and execution timeout of a VIT. Let R denote the reward and D refers to the initial deposit demanded. Then $R = D = k_1 \cdot \eta$, while its execution timeout $\delta_e = T_e \cdot \delta_p$, where $T_e = \lceil k_2 \cdot \eta / \delta_p \rceil$, ceiling in average duration of a period. k_1 and k_2 are community-defined constants which remain steady in a relatively long time. We assume they are reasonably setup, which make R attractive to most rational disinterests[12], and make δ_e adequate for them to independently finish the execution w.h.p. User is required to associate sufficient deposit and reward according to η, excessive fees will be refunded faithfully after validation. Due to the same initial yield, VITs are treated indiscriminately. Nondeterministic incentive risks are extracted from Lever.

Combined with the models above, Lever drastically decouples intensive overhead from classical Byzantine consensus, hence significantly mitigates threats from lethal nondeterministic elements and stays compatible to previous works. Below, we prove the soundness of such decoupling and demonstrate the effectiveness of our hardness model in bounding the actual workload of validation.

Theorem 1 *Any validator in Lever could predicate the validity of a transaction carrying reward $k_1 \cdot \eta$ within the execution cost C, where $C < k_1 \cdot \eta$, regardless of whether there is a misstatement in hardness.*

Proof *If adversary attempts to cover up the actual hardness η_a with η, let η_x denote min(η_a, η). In both channels, validator could terminate validation after finishing operations weighted η_x, and directly mark the transaction as invalid. Since $C/k_1 < \eta_x \leq \eta$ always holds, validator will never undertake workload:* 1) *with insufficient incentive,* 2) *of which hardness exceeds the limit of current channel. Therefore, the decoupling is complete with a stable hardness boundary η'.*

B. Data Model

We inherit and modify the elegant data model of Chainspace [10] with above mechanisms. In Lever, an object o acts as the minimum functional atom which holds state, while a smart contract is decomposed into multiple objects which greatly promote the concurrency of system. Objects are endowed with cryptographically derived unique identifiers. Valid transactions consume the input objects $\vec{w}$ and activate the output objects $\vec{x}$, thus conduct the transition of states.

A contract object o_c maintains a namespace containing several objects. In each contract o_c, a procedure p is defined to accomplish the computation off-chain while a deterministic checker v is applied for on-chain validation:

$$o_c.p(\vec{w}, \vec{r}, par) \rightarrow \vec{x}, ret, \eta$$

[11] Ethereum only requires an ambiguous user-defined gasLimit in transaction, as the upper bound of complexity. This results in a loss in stake when miners refuse to refund the rest gas according to the real execution [62]. Also, it adopts a flexible but nondeterministic parameter (gasPrize) to allow custom incentive of transactions.

[12] The reward need to cover the overhead of a single validation, as well as giving a fair incentive to the validator.

$$o_c.v(\eta, p, \vec{w}, \vec{r}, par, \vec{x}, ret, dep) \rightarrow \{true, false\}$$

Where *par* and *ret* respectively denote the input data and returned parameters, $\vec{r}$ is the list of objects referenced during execution and *dep* contains necessary information which proves the correctness of execution. For ease of presentation, we summarize *par*, *ret* and *dep* as *data* and represent a regular transaction in Lever as:

$$Tx := \langle o_c, \eta, asset, \vec{w}, \vec{x}, \vec{r}, data \rangle_{\mathsf{sk}_i}$$

Therein, *asset* generally indicates the balance proofs used for deposit and reward.

Also, we create *LeverTx* as a special type of transaction to represent the proposals of validators and challengers. As a variant of ledger transfer with negligible overhead, it routes the simplified audit information and *asset* to the Daily Channel and then gets committed to update the latest state of validation. Accountable evidence is thus left for incentive settlement. Similar to Segregated Witness [63], the transaction set is prunable after the VIT achieves its finality, which means no increasing pressure on storage is included.

Let *txid* denote the unique identifier of its target VIT. *vd* refers to the binary verdict given by the proposer. The cumulative number *ph* shows the current stage of game and can be parsed to a round-stage pair (like Verify-3 means the proposal belongs to the Verify stage in round 3). Then *LeverTx* is formed in:

$$LeverTx := \langle txid, asset, vd, ph \rangle_{\mathsf{sk}_i}$$

Combined with the design in Lever Channel, we can decompose the intensive validation into a series of LeverTxs depicted in Figure 3.

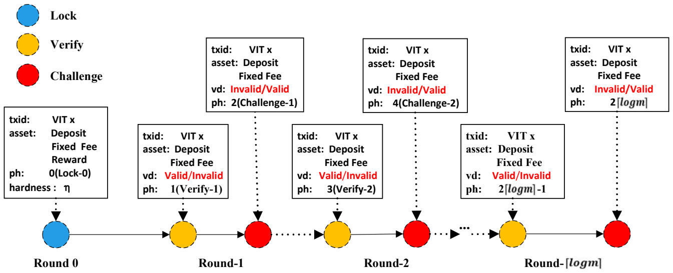


Fig. 3: LeverTxs and Decomposition of VIT

C. Lever-Boost Game

In this part, we present the Lever-Boost game, a scalable validation pattern specialized for intensive workload. The workflow is elaborated from the lifecycle of a VIT. First, the following functions are defined:

- $\mathsf{pk} \leftarrow TaskAlloc(txid, \{\mathsf{pk}\}, \mathsf{rnd})$. On input the identifier *txid* of VIT, the list of incumbent validators and a fresh random seed. The function randomly elects a validator with sufficient deposit. Note that, in Lever, validators are required to pre-store the stake for undertaking upcoming validation tasks. Specific accounts are built to transparently manage the balance according to their endorsements.
- $vd \leftarrow Backstop(txid, t, \{\mathsf{pk}\}, \{LeverTx\})$. On input *txid*, the current clock counter *t*, the validator list and the log of related LeverTxs. The function initiates the backstop protocol in Section IV-E and returns a final verdict *vd*.
- $asset \leftarrow Settlement(txid, \{LeverTx\}, vd)$. On input the final verdict and the list of LeverTxs, the function carries out settlement on the *asset* w.r.t. the incentive mechanism in Section IV-D.

The Lever-Boost game could be disassembled into four phases:

- (**Lock**) On receiving any VIT, members in Daily channel will do the underlying checks:
  - Check if *asset* is adequate according to hardness η.
  - Check whether the input and reference objects $\vec{w}, \vec{r}$ do not conflict with any proposed transactions.

  The VIT will be aborted if any check fails. Otherwise, a new $LeverTx := \langle txid, \eta, asset, \text{Lock-0} \rangle$ will be parsed and committed by DailyBFT. Meanwhile, the VIT along with $\vec{w}$ and $\vec{r}$ will be locked in the block. By next period, a fresh rnd can be retrieved from DailyEL, which helps to assign the validation task to a certain validator V. His required deposit is thereby frozen. The first round of game starts and the execution timeout is activated.
- (**Verify**) V obtains *vd* by executing the checker $o_c.v()$ and then he creates $LeverTx := \langle txid, vd, \text{Verify-}s \rangle_{\mathsf{sk}}$ where s denotes the round of game. The proposal should be broadcasted to the Daily channel within T_e periods. If he fails, the committee will forfeit his frozen stake and reselects a new validator when the timeout expires.

  On receiving LeverTx from any validator, members assure his authority on this validation and commit the LeverTx by DailyBFT. At the beginning of next period, the challenge stage starts with a timeout of T_e periods.
- (**Challenge**) Any node in the system could create a LeverTx to oppose the verdict of last verifier, which will be committed by DailyBFT, provided:
  - The associated VIT is currently in challenge phase.
  - The challenger has pledged adequate deposits.
  - Conflicting challenge proposals have not been committed in previous consensus.

  By next period, only one challenge proposal will be added to $\{LeverTx\}$[13]. A new validator will be allocated to handle the validation within T_e periods. The game will evolve into the next round, iteratively undergoes the same Verify-Challenge circle. When it finishes the $\lceil \log m \rceil$th round and still remain unsolved. The backstop protocol is employed to return the final verdict.

  In case there collects no valid proposals from challengers when the timeout expires. Verdict from the last validator will be regarded as the final verdict.

[13] It is possible when two or more valid proposals arrive at the same round of DailyBFT. Then, by next period, the fresh rnd will be used to fairly select one to be committed and abort the others.

TABLE II: Incentive $\mathcal{I}$ provided by the Lever-Boost Game at round s

Role	Deposit	Expense	Incentive-Correct	Incentive-Wrong
Founder	$2^{s-1}a$	$2^{s-1}a + fee$	$-2^{s-1}a - fee$	$-2^{s}a - fee$
Verifier	$2^{s}a$	fee	I_R[1]	$-2^{s}a - fee$
Challenger	$2^{s+1}a$	fee	I_R	$-2^{s+1}a - fee$

[1] $I_R = \begin{cases} 2^{s-1}a - fee & o_c.v(\mathsf{VIT}^{s-1}) = 1 \\ 2^{s}a - fee & o_c.v(\mathsf{VIT}^{s-1}) = 0 \end{cases}$

- (**Settlement**) The committee recursively conduct the settlement on incentive w.r.t. the final verdict. If the VIT is valid, $\vec{x}$ will be activated. Otherwise, $\vec{w}, \vec{r}$ will be unlocked.

Here we expound some vital designs in Lever-Boost game: It is noticeable that intensive workload is randomly assigned to validators, which ensures a stable and positive payoff for rational disinterest regardless of fierce competition[14], making the position of validator persistently attractive. While potential challengers from the whole system compete to publish their queries, premeditated censorship can hardly prevent valid challenge proposals to enter the block. The pattern does not impose any redundant interactions or trust foundation among potential disinterests and stakeholders, which is perfectly deployed as a backbone solution.

To increase motivation of challengers and purify the Byzantine nodes, we novelly set up a peculiar election throughout the game. Once the finality is achieved, any incumbent validator submitting the wrong verdict will be substituted by the corresponding challenger in that round. The committee will timely adjust the membership of validators by DailyEL.

D. Incentive Design and Settlement

Table II reveals the reference of settlement in Lever-Boost game at round s, where the initial deposit and reward are both $a = k_1 \cdot \eta$ and fee denotes the expenses for storage and communication required by Daily channel. We consider the validation cost $c \gg fee$ by default, and obviously $a > c + fee$. There are three roles in Lever-Boost game: transaction founder, validator and challenger. Let u_f, u_v, u_c denote their incentive respectively, which depends on the correctness of their verdicts. Proposing a wrong verdict or broadcasting invalid transactions only results in the loss of deposit, while validating honestly always brings the corresponding reward. Since the verdicts of challenger and validator contradict to each other in the same round, there is always a share of deposit from the defector to upgrade the incentive for VIT.

Such effect can be explicitly demonstrated in Figure 4: At Stage I of the game, surplus incentive can be calculated as $-(u_f + u_v + u_c)$. In case when challenger makes the wrong verdict, an extra reward of $4a$ is generated[15]. While this value remains $2a$ in honest cases, it totally covers up the incentive requirements for twice executions. To clarify such difference, we define VIT^s as the encapsulated validation task in round s, of which validity is determined by the correctness of its challenge proposal.

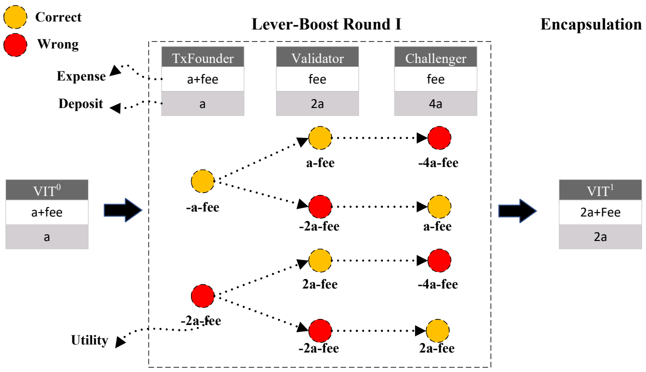


Fig. 4: Incentive of Lever-Boost Game at Round I

To generalize, every complete round of Lever-Boost game will double the incentive of VIT. The reward $(2^{s-1}a - c)$ grows exponentially as the game goes on, making the VIT itself a healthy goose egg, which attracts increasing rational disinterests to join the competition in challenge periods. They will become stakeholders of the transaction after handling the workload, henceforth fighting against stubborn adversaries as well as chasing for the massive reward. Once Lever-Boost finishes its $\lceil \log(m) \rceil$th round, it accumulates enough incentive to obtain the correct verdict from backstop protocol. As a result, even in the worst case, the game can rapidly converge to finality. Balance of such conflict will always tend to justice.

Remark. Combined with the Lever-Boost game, the following improvements have been equipped comparing to classical challenge-response patterns:

- The maintenance of validity is no longer altruistic. Every honest validator and challenger could receive a compatible reward, which stays linear with his deposit.
- The challenge interface achieves perfect resistance on abuse. Even an adversary with unlimited asset is unable to prevent the finality or censor the validation.
- Interactive requirement for stakeholders has been greatly released. The procedure is well-simplified, adaptive to capacity of adversary and bounded within $2\lceil \log m \rceil T_e$ periods.
- A friendly deposit is employed throughout the game, based on the cost of a single validation. The scheme is cost-efficient for user as the increasing reward only comes from the deposit of adversary.

We provide detailed analyses and proofs in section VI-B.

E. Finality Game

To prevent the Lever-Boost game from falling into an endless asset campaign, we design the Finality game to accomplish duplicated intensive validation under the BR assumption, which serves as a solid guarantee for validity and finality of the framework. The game is deployed in a Sybil-resistant committee where m incumbent members (termed as judges) are randomly selected to form a dispute group. They should

[14] We discuss the imbalance brought by intact competition in Appendix B.

[15] Note that, *fee*s are extracted as commissions of daily Byzantine consensus.

reach consensus on the validity of VIT within finite periods. While at least m shares of incentive have been accumulated for dispute resolution[16], the prior challenge is to eliminate freeloading, which could totally break the independence of validation and fairness in reward allocation. Also, a proper threshold Th should be established to adjudicate the validity of result. Hence, we build the following three-phase protocol (illustrated in Figure 5) to meet such targets:

Let a_r be the expected reward for each node and d_r denotes the required deposits w.r.t. hardness η.

- (**Lock**) On receiving a VIT which triggers the dispute. In Lever channel, a container $ct := \langle txid, \eta, t, \{\mathsf{pk}\}, \{Comm\}\rangle$ will be constructed for anchoring the audit states of validation, where $txid$ and η respectively denotes the identifier and hardness of VIT, t refers to the current clock counter. All incumbent judges with sufficient deposits will be automatically recorded to the list $\{\mathsf{pk}\}$[17] with d_r frozen on their accounts, and $\{Comm\}$ is initialized to gather the verdicts from them. Meanwhile a timeout of T_e periods will be activated.
- (**Commit**) Any recorded judge is required to finish the validation and obtain the *verdict*. Then he generates a random seed r and make commitment with a random oracle: $Comm := \mathcal{H}(verdict||\mathsf{pk}_i||r)$, and broadcasts it with the identifier of container $\langle Comm, ct\rangle_{\mathsf{sk}_i}$ within the execution timeout. Valid commitments will be updated in $\{Comm\}$ through DailyBFT.
- (**Open**) Once all relevant commitments have been collected or the timeout expires, all related judges need to reveal their commitments by broadcasting $\langle verdict, r\rangle_{\mathsf{sk}_i}$ within one period. The committee will allocate incentives based on the statistical results over judges' votes:
  - If any of the verdicts $\{valid, invalid\}$ get enough votes exceeding Th, it will be taken as the validity of VIT, and the container will be closed. Judgers who has made the correct choice will unfreeze their deposits and gain the reward a_r, while others will lose their deposits.
  - If none of the verdicts collects enough votes, all involved judges will temporarily lose their deposits. The container will be suspended. After m periods, a new container will be constructed to resolve the same dispute. The game will be cycled until the correct verdict stands out. Then restricted punishment could be applied to the suspended containers, where honest judges could unfreeze their deposits according to their votes. In addition, they could also obtain the deserved reward a_r by consuming the forfeit of defectors.

Note that, any equivocal or aborted commitment will leave accountable evidences, making a judge directly lose his deposit, and deliberate censorship could only result in a view change to the malicious leader.

In section VI-A, we carry out detailed game-theoretic over the game and proves it as a self-contained solution for the Verifier's Dilemma.

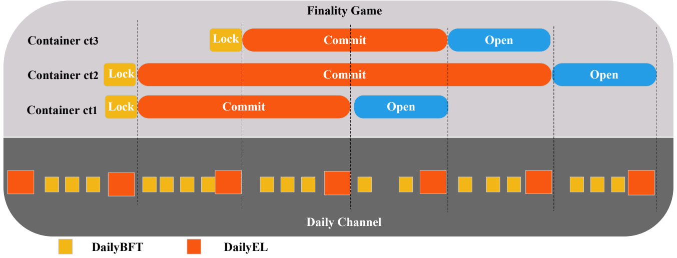


Fig. 5: Finality Game

V. Sharding-based Instantiation

A. System, Role and Responsibility

We build Lever on top of existing sharding-based architecture [7], [8], [10] to pursue perfect scalability over any conceivable workload. Depicted in Figure 6, a unique root committee C_R is set up and a global clock is built according to its election consensus. In each period, k sharding committees C_S run in parallel to take the clear majority overhead of validation. Like Rapidchain [7], there exists $m = c \log n$ nodes in each committee, where c is a constant determined by the secure parameter. C_R efficiently arranges the election of validators in C_S via a multiple-rolling manner [40]. We denote by an *epoch* as the fixed time between global reconfiguration of C_S. In practice, we assume $e \gg m \cdot \delta_p$.

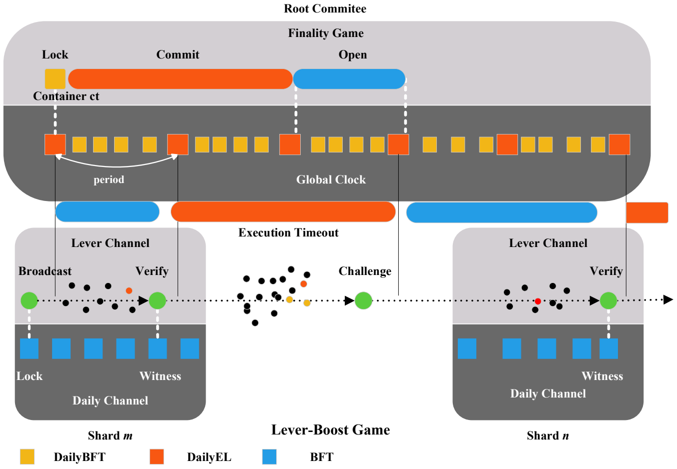


Fig. 6: Sharding-based Instantiation of Lever

Concretely, as the pivot of system, C_R could employ most committee-based consensus with perfect liquidity, such as Solida. While valuable mining reward attracts judges with significant computation power to join the committee, it takes on greater responsibilities of incentive management, task allocation and dispute resolution. By timely tracing the audit states according to valid LeverTxs, it also provides a fast and consistent view on the finality of VITs.

Whereas shard committees efficiently undertake burdensome communication and storage over contract data, programs

[16] At least $2m$ shares when $\mathsf{VIT}^{\lceil \log m \rceil}$ is false.

[17] If the number of eligible judges is less than the prescribed scale, the construction will be postponed to subsequent periods.

and states. In each period, they asynchronously achieve several rounds of classical Byzantine consensus to directly handle transactions from external ecology. Also, C_S timely checks and anchors proposals from its subordinate validators. When cross-shard operation is needed, Inter-Committee Routing Scheme [7] is employed to route necessary data among shards.

Below, we elaborate the principle of such work separation and import a series of optimizations to streamline our framework.

B. Optimizations

1) Election: In Lever, deposit is the vital certificate for intensive validation. We assume a well-proportioned distribution over nodes' stakes and elect validators through a hybrid variant of PoS and PoW:

At the beginning of each epoch, by employing protocols of distributed randomness generation [64]–[66], C_R provides an unbiased random seed $seed_e$ to initiate the election of validators. Where candidates are required to generate and broadcast the election certificates within the current epoch:

$$cert = \mathcal{H}(seed_e||\mathsf{pk}||lockProof||nonce) \cdot f_c(deposit)$$

Therein, $\mathcal{H}$ is a random oracle and $f_c(\ )$ is a deterministic convert function with strict monotonicity. *nonce* denotes the optimum value tested out to attain the smallest *certificate*. Inspired by Tendermint [67], candidate could lock some stake in C_R as his *deposit* to alleviate the difficulty in generating a smaller $cert$. The stake could be retrieved once he lost the election and $lockProof$ denotes the valid balance proof of such act.

C_R anchors the valid certificates throughout the epoch. At the beginning of next epoch, top-ranking candidates are unbiasedly assigned to C_S with reference to secure reconfiguration mechanisms [7], [8], where there exists less than $m/3$ Byzantine nodes in each shard with high probability.

The mechanism makes comprehensive requirements on execution power and stake of potential validators, which intuitively ensures better capacity of system in intensive validation.

2) Coordination on Workload: To balance the resources and workload over various shards as well as reducing the occurrence of censorship, the framework imposes little conflict of interest between judges and validators. C_R is therefore competent in making fair and uniform workload coordination w.r.t. rnd provided by its DailyBFT.

Requests for contract registration are randomly assigned to shard committees. The upcoming transactions which target the contact will be routed to the certain shard, where subordinate validators are responsible for handling the validation and tracing the states of registered contracts. In dealing with any VIT undergoing more than one round of game, the validation task is then assigned randomly among all existing shards. Although this incurs some inter-shard operations, it could make the adversary's censorship worthless when targetedly corrupting any shard. Also, the pressure of validation gets more balanced through the design.

3) Incentive Management: If each shard autonomously manages its *asset* for endorsement, then a VIT with r rounds of Lever-Boost game could entail at most $2r + 1$ cross-shard transactions on settlement, which incurs intolerable latency on validation. To address the issue, C_R acts as an incentive beacon to collect *asset* from valid LeverTxs and achieves a timely and trustworthy settlement once the VIT obtains its finality.

Figure 7 illustrates the states preserved by C_R. For incumbent validators and judges who have their deposits prestored in C_R, specific accounts are set up to keep track of the endorsement events in their linked committees. Deposit has two states: it gets *frozen* when used to handle the validation of VIT, and turns into *dynamic* when the final verdict confirms the correctness of endorsement. Otherwise, the owner would directly lose the part of stake. Note that, the dynamic deposit can only be applied for validation before its owner has been rolled out of his linked committee, while the frozen part can only be released until all his endorsed VITs get finalized. Incumbent validators can replenish their deposits to C_R at any time during their lifecycles, any incentive he collects during lifecycle will also automatically add to the balance of deposit. Since each account has a unique linked shard supported for capital flows, the risk of double-spending on deposit is totally eliminated. The mechanism provides a transparent and quantitative view on the capacity of validators by using deposit as an accurate indicator.

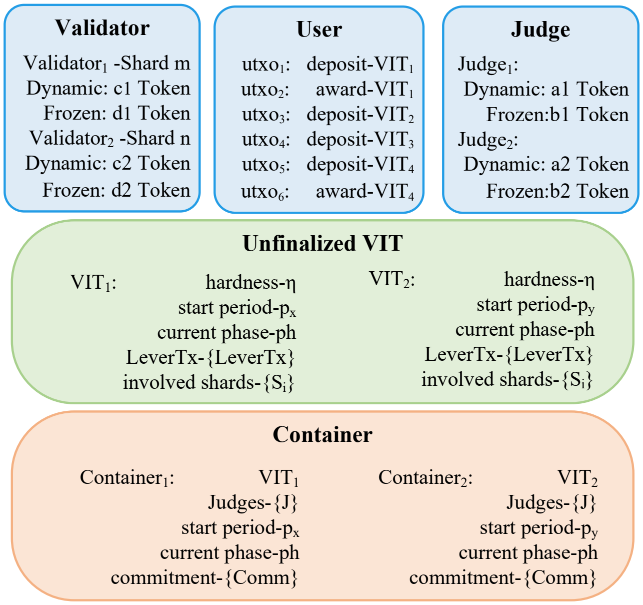


Fig. 7: Storage of States in C_R

For user, possibly as a transaction founder or a challenger, their mortgaged *asset*s are managed in the form of UTXO (unspent transaction output), linked with its incoming shard. The *asset* will be unlocked equitably after the settlement. C_R also preserves the simplified audit state of all unfinalized VITs and unclosed containers, which are devoted to asserting the finality of VITs.

4) Witness and Batching: In Lever, C_S runs a faster Byzantine consensus to perform witness and pre-processing on massive proposals from user and validators. In each round of consensus, after finishing routine checks, it batches the valid LeverTxs, generates the corresponding Merkle proof and routes them to C_R along with at least $\lceil 1/3 \cdot m + 1 \rceil$ valid signatures. Here, LeverTx is viewed as a typical cross-shard ledger transfer, which creates negligible overhead.

In C_R, judges collect all the untreated batches from C_S, check the Merkle proof and signatures. When a newly-spawned judge appears from DailyEL, C_R firstly performs task allocation with rnd and updates the state of assets and VITs according to the coordination results. Then it extracts pairs of finalized VITs with their verdicts $\{(txid, vd)\}$ w.r.t. the global clock t and the state of containers. Judges accordingly complete the settlement on *asset* and update the configuration $\{\mathsf{pk}\}$ of shards to purify Byzantine validators. Along with the daily election, a landmark block of last period could be finally constructed in: $Block^t := \langle t, \{\mathsf{pk}\}, \{(txid, vd)\}, \{LeverTx\}, \mathsf{rnd}\rangle$.

In each period, every validator would synchronize the finality of VIT and deposit balances correlated to his shard, and accordingly update the state of objects. Integrated protocol of our instantiation is formalized in Figure 20.

Remark: The optimizations above achieve secure, consistent and effective management on incentive and finality of Lever-Boost game. Hierarchical election with staking is leveraged to filter competent validators while maximizing the throughput of system. Tracing the states of LeverTxs in a uniform manner avoids intricate cross-shard operations. Optimized latency and communication complexity are thus achieved while maintaining the fairness and orderliness in workload coordination. Whereas C_S greatly reduces the workload of root committee through witness and batching, the risks from DoS attack could be totally erased. Scalable intensive validation is well achieved in computation, storage and communication. Even in the worst-case setting, a VIT could be finalized within $\lceil \log \log n \rceil$ rounds where at most $O(m + \log m) = O(\log n)$ times of validation is taken.

C. Achieving Atomicity from S-BAC

We achieve atomicity on Lever with the help of Sharded Byzantine Atomic Commit (S-BAC) protocol invented by Chainspace [10]. S-BAC incorporates Byzantine agreement with the two-phase commit protocol, enabling each involved shard validate independently and conduct replicated Byzantine consensus on cross-shard transactions. Finally, shards exchange decisions and uniformly accept the transaction only when every shard has committed it.

We employ S-BAC to attain a consistent Lock phase for the Lever-Boost game. Following the workflow in Figure 8, upon receiving a cross-shard VIT, every input shard independently does the routine checks and commits to lock or abort the transaction. Then if no abort exists, the LeverTx is collectively committed by the shards and routed to C_R, where judges randomly choose a certain validator in input shards to undertake the validation. Thus, the single validation starts with a unique route. When finality is achieved, related shards carry out atomic transitions on objects according to the final verdict.

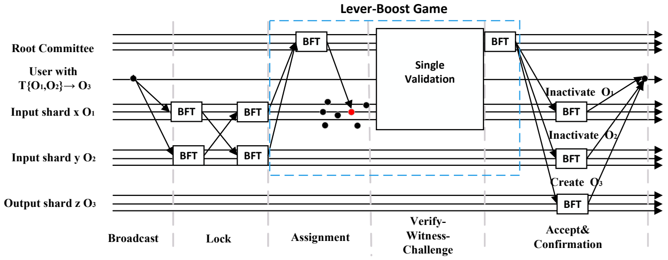


Fig. 8: Atomic Inter-shard Validation

D. Flexible Interface to Off-chain Ecology

To facilitate mutual promotion between Lever and off-chain techniques, we propose the construction of Private Shard, a flexible interface specified for the off-chain ecology. Private shards could act as a sandbox for off-chain protocols such as State Channel [12], Arbitrum [11], SmartCast [17] etc. triggering frequent events and realizing high privacy standards. We elaborate its workflow by defining the underlying functions:

- $P_S \leftarrow Reg(\{M_{\mathsf{pk}}\}, checkers, o_i)$. By sending a registration request to any C_S, nodes could bind customized membership $\{M_{\mathsf{pk}}\}$, the initial state o_i and a prescribed set of checkers to a private shard P_S. An authorization list is set up to drive a signature oracle, taking in charge of state evolution.
- $Update(o_f, \mathsf{Sig}, h, cnt)$. Any member in $\{M_{\mathsf{pk}}\}$ could update the latest state of private shard o_f by providing a digest h to protect the integrity of off-chain operations and a complete collection of signatures on $\langle o_f, h, cnt\rangle$. An incremental counter cnt is included to prevent reply attacks.
- $o_f \leftarrow Dispute(VIT, h)$. When there appear operations importing the matters of agreement failure, public competition and intense conflict of interests. Any member in the private shard could update the agreed states of P_S and deliver the workload to Lever with the expense of a single validation. Stakeholders could propose valid VITs to drive the state transition in a fair and non-interactive manner. Then members could receive the final judgement on states and vote to continue the procedure in private shards.

To summarize, Lever extends the applicability of off-chain protocols and greatly reduces the cost and latency on potential dispute resolution.

VI. Security Analysis

A. Security of Finality Game

We first notarize feasibility and ensure specific parameters of the Finality game, underlying analyses are made from the game-theoretic view:

Roles and Strategies. There could be three types of player in the Finality game: For rational judges, a cooperator always proposes the correct verdict by validating honestly. Whereas a defector will not adopt aggressive strategies like deliberately broadcasting the wrong verdict after validation, and since any attempt to abort will result in the worst payoff, staying lazy and guessing obviously becomes his unique dominant strategy. For Byzantine players, we consider the worst situation that they are all controlled by one adversary who knows the verdict in advance. They will uniformly vote for the wrong verdict, rather than launching forceless abort attacks.

Formalization. We formalize the Finality game as a Bayesian Machine game $\mathcal{G} := ([m], T, P_r, X, \mathcal{M}, u)$, where $[m]$ denotes the set of players, T refers to their types discussed above, P_r is a distribution on T, and $X = \{correct, wrong\}$ is the possible action sets of the game. u refers to the utility function of the game, determined by the votes over actions. $\mathcal{M}$ is the set of possible Turing machines for players to obtain the verdict. Cooperators utilize a uniform machine to consume c and output the *correct* result, while the defectors' machine takes trivial cost to output a *random* verdict. Since defectors cannot retrieve any information about votes before the *Open* phase, and they are not credulous to any unreliable sources out of rationality, the output of their machines can be fitted by a Binomial distribution with $1/2$ as its probability of correct.

To simplify the analysis, we ignore the trivial *fee* generated from communication and storage. The workflow of our game is similar to threshold public goods game [57], of which researches [27], [28], [56], [58] provide great inspirations to analysis and refine our game. However, as validation is set as the unique signal for rational players to facilitate unanimous actions, our game can be rescued from falling into the scope of Social Dilemma [51].

Guarantee for Validity. The machines $\mathcal{M}$ transfer certain P_r on T to deterministic distributions on X, which could be categorized to three situations that directly determine the utilities of players and the validity of $\mathcal{G}$: `JusticeWin` and `EvilWin` respectively describe when *correct* or *wrong* verdict collects votes over Th, otherwise `NoOneWin` occurs where every judge lose his deposit.

Let $f(k, n, p) = \Pr(X = k) = \binom{n}{k} p^k (1-p)^{n-k}$ denote the probability mass function of Binomial distribution and $F(k; n, p) = Pr(X \leq k) = \sum_{i=0}^{\lfloor k \rfloor} f(i, n, p)$ refer to its cumulative distribution function. We transfer P_r as the type profile of $\mathcal{G}$, where k cooperators, i defectors and t Byzantine players attend the game[18], the probability of occurrence on above situations can calculated as:

$$p_J = \begin{cases} 1 & k \geq Th \\ 1 - F(Th - k; i, 0.5) & 0 < k < Th \end{cases}$$

$$p_E = \begin{cases} 1 & t \geq Th \\ 1 - F(Th - t; i, 0.5) & 0 < t < Th \end{cases}$$

$$p_N = 1 - p_E - p_J$$

To guarantee the validity of $\mathcal{G}$, we only need to restrain the probability of `EvilWin` to negligible under the worst case configuration $(t = \lfloor 1/4m \rfloor, i = m - t)$, thus Th needs to satisfy the following condition:

$$1 - F(Th - \lfloor 1/4m \rfloor; \lceil 3/4m \rceil, 0.5) < 10^{-\lambda} \qquad (1)$$

TABLE III: Utility Expectation with any fixed type profile $(k, i, t) - (Cooperator, Defector, Byzantine)$

Situation / Type	`JusticeWin`	`EvilWin`	`NoOneWin`
Cooperator	$(a_r - c) \cdot p_J$	$-(d_r + c) \cdot p_E$	$-(d_r + c) \cdot p_N$
Defector	E_J[1]	E_E[2]	$-d_r \cdot p_N$
Byzantine	$-d_r \cdot p_J$	$a_r \cdot p_E$	$-d_r \cdot p_N$

[1] $$E_J = \begin{cases} \sum_{j=0}^{i} \frac{a_r j - d_r (i-j)}{i} \cdot f(j, i, 0.5) & k \geq Th \\ \sum_{j=Th-k}^{i} \frac{a_r j - d_r (i-j)}{i} \cdot f(j, i, 0.5) & 0 < k < Th \end{cases}$$

[2] $$E_E = \begin{cases} \sum_{j=0}^{i} \frac{a_r (i-j) - d_r j}{i} \cdot f(i - j, i, 0.5) & t \geq Th \\ \sum_{j=Th-t}^{i} \frac{a_r (i-j) - d_r j}{i} \cdot f(i - j, i, 0.5) & 0 < t < Th \end{cases}$$

Under an appropriate Th, we can show the changes of probabilities over all possible P_r with $\lfloor 1/4m \rfloor$ Byzantine nodes in Figure 9.

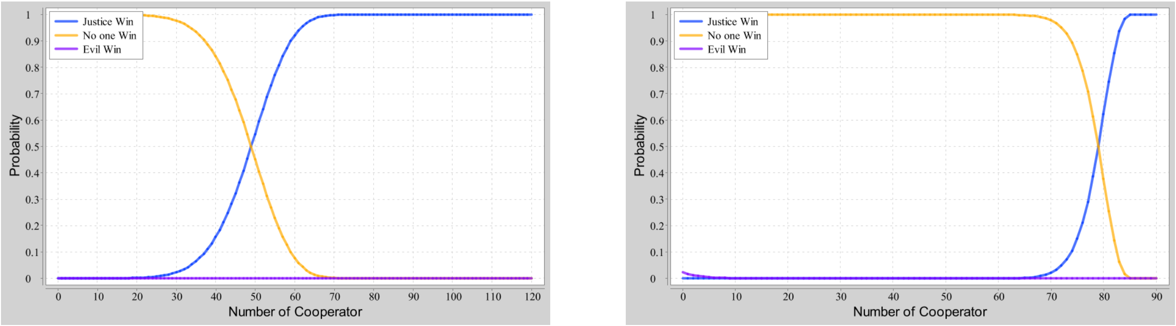
Calculation of Utility and Equilibrium. On this basis, we first analyze $\mathcal{G}$ in an one-shot game without considering the effect of restricted punishment. Given any P_r, we could calculate the fractional utility expectation $E_{Pr}(u_i^{\mathcal{G}, M})$ under every possible situation (shown in Table III), then carry out the summation to get the expected utility of each role U_i. By rotating on $i \in [1, m-t]$, we can show the changes of U_i on rational behaviors in Figure 10. Apparently, for $0 \leq t \leq \lfloor 1/4m \rfloor$, there always exist two pure-strategy t-immune equilibriums in $\mathcal{G}$: a Pareto-efficient equilibrium σ_1 appears when all rational judges cooperate and acquire deserved rewards a_r rather than guessing. Whereas a misshapen equilibrium σ_2 arises when everyone defects and loses his deposit, defectors suffer a lesser loss than cooperators by saving the validation cost.

Selection of Equilibrium. We follow the Harsanyi's theories on cherry-picking the advantageous equilibrium. Under the premise of payoff-dominance, σ_1 is undoubtedly superior to σ_2. Thus, $\mathcal{G}$ could be terminated within $T_e + 1$ periods with high probability.

However, things become complicated under the risk-dominant assumption, where rational players hold intense uncertainty and afraid to lose more. To evaluate the success rate [27] (emerge probability) of σ_1, we need to calculate the mixed strategy Nash equilibrium σ_3 of the game, where every rational player cooperates with probability p_m, defects with $1 - p_m$, making utility expectation indifferent between cooperation or defection. The profile of type can be derived as $((m-t) \cdot p_m, (m-t)(1-p_m), t)$. Combined with the relations in Table III, σ_3 can be found by solving the underlying equation about p_m employing the method of dichotomy:

$$(a_r - c) \cdot p_J - (d_r + c) \cdot (p_E + p_N) = E_J + E_E - d_r \cdot p_N$$

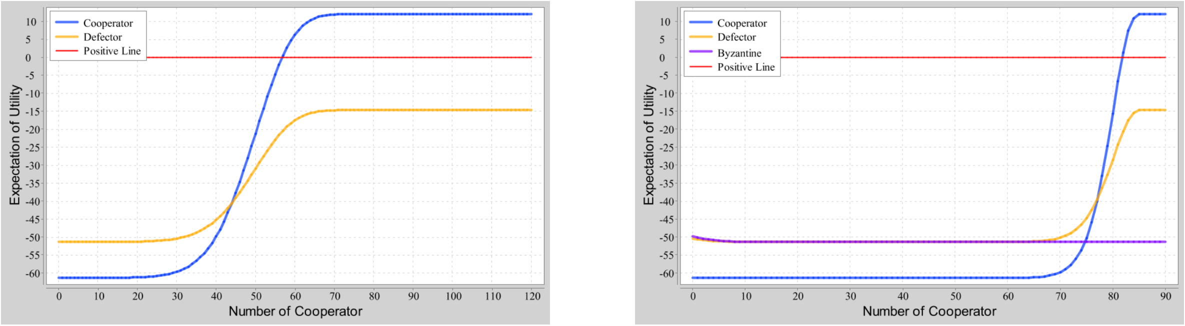
[18] $k + i + t = m, t \leq \lfloor 1/4m \rfloor$



(a)

(b)

Fig. 9: Probability of all situations ($m = 120, Th = 84$) under (a) Rational case, (b) Worst-case BR assumption ($t = 30$).



(a)

(b)

Fig. 10: Utility expectation ($m = 120, Th = 84, c = 10, a_r = 22, d_r = 51$) under ($a$) Rational case, ($b$) Worst-case BR assumption.

The success rate of σ_1 can be calculated as $S_r = 1 - p_m$ and the dispute could be resolved within $1/S_r$ rounds of repetitive game[19] on average. It is clearly observed from Figure 11 that the finality of game can be achieved within finite games. However, as the ratio of Byzantine increases, there produces more latency on the finality of VIT, creating considerable liquidity pressure on deposits of judges.

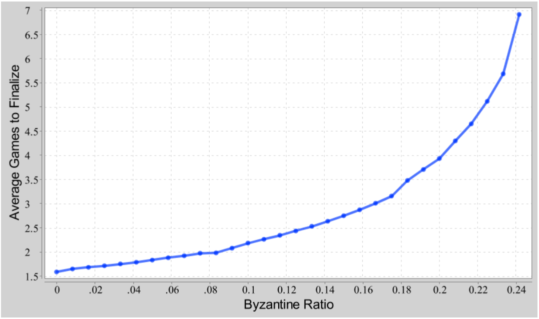


Fig. 11: Average games to finalize without restricted punishment under risk-dominant assumption ($m = 120, Th = 84$)

Effect of Restricted Punishment. Now we consider the refinement brought by restricted punishment. Once $\mathcal{G}$ is finalized, for every relevant suspended container, cooperative judges could unfreeze their deposits according to the correct verdict. In addition, they could also obtain the reward a_r by consuming the deposits of defective players. To deploy such mechanism, d_r is set to meet the following condition:

$$a_r \cdot Th \leq d_r(m - Th) \Rightarrow d_r \geq \frac{Th}{m - Th} a_r \tag{2}$$

With gratuitous penalties refunded, the adjusted utility expectation of $\mathcal{G}$ can be shown in Figure 12. Cooperation is the unique dominant strategy of the game, providing a stable positive expectation of $a_r - c$, whereas defector and Byzantine players always get negative payoffs of $1/2 \cdot (a_r - d_r)$ and $-d_r$. Obviously, the underlying theorem could be derived:

Theorem 2 *The Finality game is* DSIC *and finalizes with the correct verdict within $T_e + 1$ periods w.h.p.*

Recall every judge primarily has his mining reward W as deposit, and R denotes initial reward provided by transaction founder. We can further infer the upper bound on hardness of transactions which Lever can safely handle:

$$\eta_{max} = \frac{R_{max}}{k_1} = \frac{W}{k_1} \cdot \frac{a_r}{d_r} = \frac{W}{k_1} \cdot \frac{m - Th}{Th} \tag{3}$$

[19] We assume the pendent votes of prior suspended containers conduct negligible interference to the forthcoming game, since: 1) All correlated judges have been rotated out of C_R w.h.p. 2) Rational players lack faith in any unreliable sources about the verdict before validation.

In Lever, since k_1 is configured under the standard of single validation, which is small enough to make the reward appropriately cover the actual validation cost. The complexity limit of the backbone is significantly raised comparing to the existing frameworks.

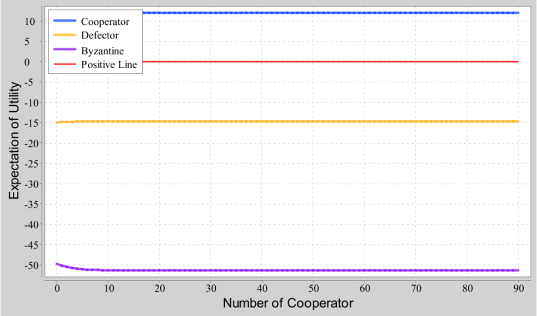


Fig. 12: Worst-case utility expectation ($m = 120, Th = 84, c = 10, a_r = 22, d_r = 51, t = 30$) with restricted punishment

B. Security of Lever-Boost Game

Referring to Theorem 1, the workload of Daily channel is safely controlled as negligible, liveness and safety is achieved under the BA assumption. Given δ_{bft} as the liveness parameter of underlying Byzantine consensus. We could further infer the finality of the Lever:

Theorem 3 *Each VIT achieves its finality w.h.p. within* $\delta_f = \delta_{bft} + (2\lceil\log m\rceil + 1) \cdot \delta_e + 3\delta_p$.

Proof *On receiving a VIT from user, relevant* C_S *take at most* $\delta_{bft} + \delta_p$ *to accomplish the Lock phase and wait no more than* δ_p *to be recorded by* C_R*. Recall the Lever-Boost Game takes at most* $\lceil\log m\rceil$ *rounds, each round could end within* $2\delta_e$*, and the Finality Game terminates w.h.p. within* $\delta_e + \delta_p$*. By the end, it takes at most* $\Delta < \delta_p$ *for* C_S *to update the states of related objects.*

Note that, any valid LeverTx could be committed by C_R within $2\delta_{bft}$. Since $2\delta_{bft} < \delta_p < \delta_e$, the liveness parameter is counted in the execution timeout, which does not affect δ_f.

There could be various strategies to deviate from the Lever-Boost game. For incumbent validators, Byzantine nodes could abort or take bribes to return the wrong verdicts, while lazy rational nodes could skip validation and randomly guess the verdict. Meanwhile, adversaries outside the committees could also make challenge proposals to disturb the game, which we define as below:

Definition 1 *External Stubborn Adversary. Potential malicious nodes who exhaust their budgets to propose challenges whenever target VITs receive correct verdicts in Verify phase of the game.*

Analyses in Section VI-A proved that the Finality game always returns the correct verdict. So long as rational stakeholders of VIT keeps challenging wrong verdicts in Lever-Boost Game, they could avoid any lost, meanwhile obtain attractive incentive from malicious nodes. Since there exists at least one rational stakeholder throughout the game, wrong proposals will never survive to become the final verdict.

We could further calculate the utility expectation of nodes according to their strategies[20]. Let p_v denote the actual emerge probability of correct VIT^{s-1}. With reference to the incentive relations in Table II, honest validator or challenger in round s respectively ends up with deterministic positive rewards:

$$u_{validator} = (p_v \cdot 2^{s-1} + (1 - p_v) \cdot 2^s) \cdot a - c \geq a - c > 0$$

$$u_{challenger} = (p_v \cdot 2^{s-1} + (1 - p_v) \cdot 2^s) \cdot a \geq a > 0$$

Whereas powerful deviation strategies including guessing, taking bribe, abort and stubbornly abusing the challenge interface only leave nodes negative expectations[21]:

$$u_{guess} = 1/2 \cdot p_v \cdot (2^{s-1} - 2^s)a \leq -p_v \cdot a/2 < 0$$

$$u_{bribe} = u_{abort} = -2^s a \leq -2a < 0$$

$$u_{stubborn} = -2^{s+1} a \leq -4a < 0$$

Apparently, validating honestly and insisting on the correct verdict is the unique dominant strategy in Lever-Boost game. Combined with Theorem 2, intuitively, we could make the following argument:

Theorem 4 *Lever achieves validity and DSIC under BAR assumption tolerance of at most* $n/4$ *Byzantine participants.*

Even in an extreme case when the honest stakeholders of a VIT have their *asset*s exhausted in the game, failed to propose a challenge. Due to the randomness of task assignment, the adversary only has a probability less than $1/4$ to launch censorship through bribing incumbent Byzantine nodes. Otherwise, in each round, the rational validator will become a firm challenger to safeguard his correct verdict. As the game goes on, the honest party will have at least a probability of $1 - (1/4)^s$ to receive such support, making the game keep its validity w.h.p. This also releases the stake requirements on stakeholders.

C. Resistance to Possible Attacks

Recall the variety of attacks mentioned in Section II-A2 which could introduce catastrophic consequences to existing frameworks, we analyze their effects on Lever and prove the robustness of our protocol at the worst-case configuration:

- **Abuse on Challenge Interface.** Other than the interactive challenge-response schemes deployed in [11], [13], [14], where external stubborn adversaries could infinitely delay the validation by making excessive challenges. There exist at most $\lceil\log m\rceil$ challenge proposals for a VIT to reach finality, setting up a fixed upper bound on latency. Furthermore, launching such attack is much more expensive

[20] *Fee* for Daily channel is omitted here on account of its trivial value.

[21] Misbehaviors of bribing and guessing also make node lose his position as a validator, extra fixed forfeits could be designed as further punishment.

and unpractical. To delay a VIT into the Finality game, adversary has to possess and burn budget of at least: $2/3 \cdot \sum_{s=1}^{\lceil \log m \rceil}(2^{s+1}a + fee)$. This even excludes the expense to bribe Byzantine validators.

- **Freeloading.** In Finality game, freeloading is eradicated with the help of commitment scheme. While in Lever-Boost game, since validation is coordinated by random assignment, the only chance occurs when competing with other challengers. However, existing proposals reveal unreliable information about the validity of transactions, and the unique winner should take full responsibility for his endorsement. Freeloading is considerably risky and makes little sense to rational nodes.
- **Censorship.** In Lever, if adversary tries to abort any valid verify proposal by performing censorship, he should control certain shard committee for at least T_e periods, which happens with a negligible probability of $p_{ce} = (\frac{1}{3})^{T_e \cdot \delta_p / \delta_r}$, where δ_r refers to the average consensus interval in C_S. As challenge proposals could be accepted by any shard, recall k denotes the total number of C_S, the probability to resist all challenge proposals is p_{ce}^k, which is also impossible. As for the root committee, any obvious censorship could only result in the mining winner rejected by the committee and completely waste his power and reward.
- **DoS Attack.** Adversary may attempt to exhaust deposit balance of certain shard by flooding vast transactions. However, since the contract registration and validation workload are coordinated randomly and uniformly among all shard committees, such attack is costly and takes trivial effect.

VII. Evaluation

A. Experimental Setup

We have implemented a proof-of-concept prototype to evaluate Lever under more realistic scenarios. Rationality is well-respected and reasonable random distributions are introduced to simulate the heterogeneity on transactions and capacity of nodes. Specifically, uniform distribution is employed to bootstrap the hardness and validity of transactions as well as the power of validators, while Pareto distribution [68] is used to initialize the deposits of validators and budgets of external adversaries. Series of optimizations have been made to abstract the burdensome computation and storage in Lever, which import trivial influence on results, while making the experiments support larger test scale and provide more generic conclusions.

As shown in Figure 13, our test framework consists of the Lever prototype and four auxiliary modules, which interacts to form the minimum unit of a self-contained blockchain system:

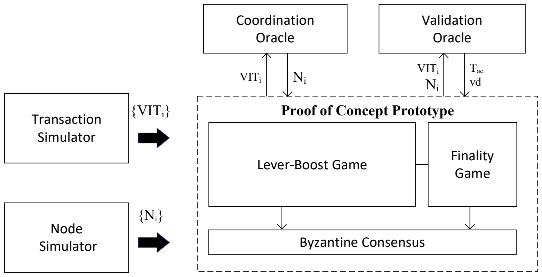


Fig. 13: Construction of Test Framework

- **Transaction simulator** continuously outputs a series of randomized VITs, of which hardness is uniformly distributed in $[1, \eta_{max}]$. η_{max} represents the complexity upper bound which Lever could safely handle (deducted in Equation 3). Each VIT has a probability of $1/2$ to be valid. As for the potential external stubborn adversary attached to it, his budget x complies with the Pareto distribution:

$$\bar{F}(x) = Pr(X > x) = \begin{cases} (\frac{x_m}{x})^{\alpha} & x \geq x_m \\ 1 & x < x_m \end{cases}$$

By setting the shape parameter $\alpha = 1$ and the minimum possible budget $x_m = 2k_1\eta$, the adversary could delay the Lever-Boost game for at least one round.
- **Node Simulator** bootstraps a certain scale of validators, whose deposits follow the Pareto distribution with minimum value $x_m = k_1 \cdot \eta_{max}/2$. Conceivable strategies are also equipped according to the type profile of game.
- **Coordination Oracle** abstracts the election of root committee, which provides a stable global clock to control the execution timeout. In each period, the oracle generates a newly-elected judge to C_R and a pseudo-random number rnd which can be used to coordinate the pending workload.
- **Validation Oracle** integrates the Turing machines w.r.t. various strategies and abstracts the execution of validation in Lever. Concretely, for rational nodes, it returns the correct verdict and a cost with reference to the hardness of VIT. In terms of Byzantine nodes, it consumes no cost for validation, but respectively returns Null or the wrong verdict to choices of abort or bribing. It returns the correct verdict at a probability of $1/2$ to those who guess the answer. A node visits the oracle with assigned VIT, then obtains his verdict, consumption and a randomized execution time T_{ac} evenly distributed in $(0, \delta_e)$. Such setting takes the heterogeneous power of validators into consideration. Also, it circumvents the burdensome computation overhead with trivial influence on evaluation results.
- **Prototype of Lever** implements the main logics of protocol in *Lever-Boost Game* and *Finality Game*. It utilizes Byzantine Agreement as the underlying module to anchor the audit state of validation, operate the overhead of storage and communication, as well as update the finality of VITs. Real-time statistics on state of nodes and transactions could be accessed from the specific data interface.

Unless otherwise noted, we deploy the following tests under the worst-case configuration, where there exists exactly $\lfloor n/4 \rfloor$ Byzantine participants in Lever, who always take bribes to propose the wrong verdict, the default test duration is set up to one epoch.

B. Performance

As for intensive validation, concrete statistical data on throughput is heavily affected by the relative connections between the deposit level of nodes and actual complexity of VITs. Without loss of generality, we evaluate the performance of Lever by setting comparative experiment with frameworks deploying the classical duplicated validation pattern, and conduct the underlying tests on same sets of randomized nodes and VITs.

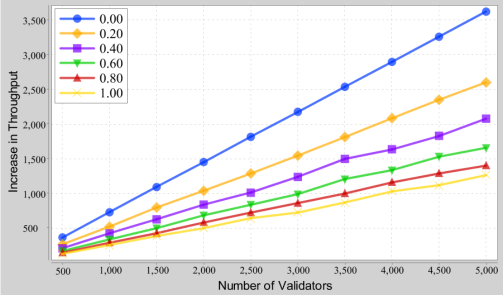


Fig. 14: Increase in throughput with different ratios of external stubborn adversary.

By consistently infusing excessive VITs to both frameworks in one epoch, we obtain the increase of throughput in Lever and distinguish the cases of varying scales of nodes and ratios of external adversary. As Figure 14 shows, Lever scales linearly in the number of validators and the effect is adaptive to the number of external adversaries. Theoretically, the increase does suffer a lower bound of $n/(\lceil\log m\rceil + 1)$, but this could be never accessed as the distribution of stakes is well-proportioned.

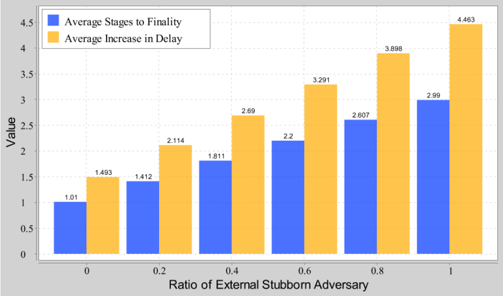


Fig. 15: Average rounds of game to achieve finality and the increase in latency introduced by Lever ($n = 1000, m = 120$).

In terms of latency, we set a fixed scale of validators $n = 1000$ with the shard size $m = 120$, and infuse $5 \cdot 10^4$ randomized VITs to the framework. For comparison, we set the latency of duplicated pattern as the execution timeout δ_e. As Figure 15 tells, a VIT could be finalized in the first round of game w.h.p. with no external stubborn adversary attached to it, while this reflects the high efficiency of Lever under the most general case. The worst-case latency is finite and affordable, adversary can averagely delay the validation for no more than 3 rounds of game.

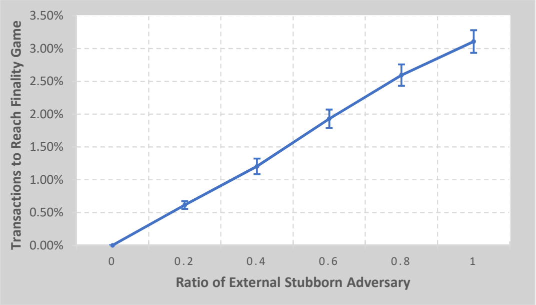


Fig. 16: Average percent of workload triggering Finality Game ($n = 1000, m = 120$).

While C_R may be viewed as a probable bottleneck of the framework, we measure the its expenses on storage and computation in above tests[22]. With 500 incoming VITs *per period* and an average execution timeout of 15 periods for each VIT, a *static* 2.802 MB workload on storage is generated owing to the prunable construction of LeverTx. Restricted by the fixed deposit of judges and the duplicated pattern, Finality game employed in C_R is inefficient, which cannot concurrently handle workload exceeds $(1+1/m)\cdot\eta_{max}$. However, we measure the average percent workload triggering the game under different ratios of external stubborn adversary by 10^5 VITs verified in Lever. As figure 16 shows, even at the worst case with extreme external pressure, over 96.5% workload of intensive validation is accomplished in Lever-Boost game, which further embodies the outstanding scalability of framework and the robustness against DoS attack.

C. Ecological Fitness

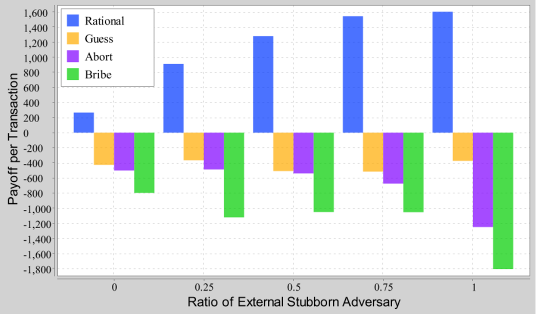


Fig. 17: Average payoff per validation for various roles ($k_1 = 10, \eta_{max} = 50, c = k_1 \cdot \eta/2, n = 1000, m = 120$).

To evaluate the effect of our framework in erasing various malicious behaviors, we build three comparison groups, which respectively equips Byzantine nodes with typical deviated strategies of guessing, abort and bribing. After handling the validation of 10^5 VITs, we measure the state of validators in Lever and illustrate each role's average validation opportunities in Table IV, and payoffs in Figure 17. As we can see, rational

[22] Since the intensive validation incurs the dominant consumption on time, trivial latency introduced by communication is ignored in the tests.

validators undertake most chances of validation and steadily obtains a positive payoff, whereas Byzantine nodes suffer substantial forfeits. The one who guesses, or bribes makes trivial influence before being ejected from his position by the corresponding rational challenger. The one who aborts the protocol exhausts his deposit and suffer maximal losses. Additionally, the emerge of stubborn adversary greatly raises the incentive of honest validation as well as the severity of punishment.

TABLE IV: Average times of validation processed by various roles in Lever.

Role / R_{st} *	Rational	Guess	Abort	Bribe
0.00	111.11	1.97	61.23	1.00
0.25	166.50	1.73	63.27	1.00
0.50	219.96	2.57	60.83	1.00
0.75	275.78	1.93	45.17	1.00
1.00	330.16	2.13	24.67	1.00

* Ratio of external stubborn adversary.

By uniformly introducing all deviated roles in one test, we gradually input small amounts of VITs to Lever and unfold the effect of the purification on Byzantine nodes. As shown in Figure 18, after verifying 3000 VITs with 1000 nodes, misbehaviors of guess and bribe become nearly extinct. Only the abort strategy survives with its waning influence owing to the exhaustion of deposits.

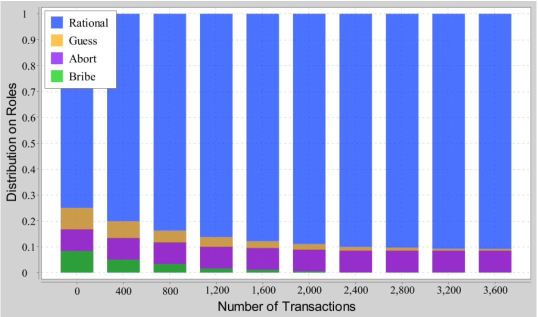


Fig. 18: Purification on potential Byzantine validators through Lever-Boost game ($n = 1000, m = 120$).

VIII. Future Work

Lever proposes a novel and efficient pattern for scalable validation, but it also leaves some limitations to address in future work. Firstly, to precisely control the execution timeout of VIT, the accuracy and stability of global clock need to be strengthened, advanced cryptographic techniques like Verified Delay Function [65] could help to make stunning improvements. Secondly, Lever relies on at least one stakeholder who has the ability to execute the verification. In reality, inspired by Pisa [54], we believe lightweight contracts attended by rational disinterests could be designed to release this assumption to only stake requirements. Additionally, it is hard to smoothly handle the boundary between VIT and general transactions, while this requires an exploration on fine-grained transition mechanisms. Finally, the fluctuation on values of digital currencies could be harmful to the robustness of Lever. An effective regulatory scheme is in need to quantify and adaptively eliminate such external financial risks.

IX. Conclusion

In this paper, we proposed the first scalable on-chain framework supportive to intensive validation, which brought about comprehensive improvements in performance, fairness and security comparing to the existing solutions. By facilitating fair competitions adaptive to the stake of adversary, we proved the pattern of single validation could be fit to deploy on backbone without any dense interaction. With ingenious deposit relationships excavated, incentive-compatible mechanisms were proposed to motivate validators according to their actual workload, breaking away from any bad dilemma on validation. In addition, lightweight measures were proposed to circumvent inherent Byzantine shackles under the sharding-based architecture.

We believe our research potentially provides useful insights into the design of incentive mechanism, Byzantine faults inhibition and ecological management in public blockchain systems. Also, it could be a powerful building block of advanced decentralized applications. We further share a series of interesting and promising prospects extended from Lever in Appendix C, waiting for academia to explore.

Acknowledgments

This work is supported by the National Key Research and Development Program of China under Grant No. 2017YFB0802500, the National Natural Science Foundation of China under Grant Nos. 61772538, 61672083, 61370190, 61532021, 61472429, and 61402029, and the National Cryptography Development Fund of China under Grant No. MMJJ20170106.

References

[1] R. Pass and E. Shi, “Hybrid consensus: Efficient consensus in the permissionless model,” in *31st International Symposium on Distributed Computing, DISC 2017, October 16-20, 2017, Vienna, Austria*, 2017, pp. 39:1–39:16. [Online]. Available: https://doi.org/10.4230/LIPIcs.DISC.2017.39

[2] I. Abraham, D. Malkhi, K. Nayak, L. Ren, and A. Spiegelman, “Solida: A blockchain protocol based on reconfigurable byzantine consensus,” in *21st International Conference on Principles of Distributed Systems, OPODIS 2017, Lisbon, Portugal, December 18-20, 2017*, 2017, pp. 25:1–25:19. [Online]. Available: https://doi.org/10.4230/LIPIcs.OPODIS.2017.25

[3] Y. Gilad, R. Hemo, S. Micali, G. Vlachos, and N. Zeldovich, “Algorand: Scaling byzantine agreements for cryptocurrencies,” in *Proceedings of the 26th Symposium on Operating Systems Principles, Shanghai, China, October 28-31, 2017*, 2017, pp. 51–68. [Online]. Available: https://doi.org/10.1145/3132747.3132757

[4] E. Kokoris-Kogias, P. Jovanovic, N. Gailly, I. Khoffi, L. Gasser, and B. Ford, “Enhancing bitcoin security and performance with strong consistency via collective signing,” in *25th USENIX Security Symposium, USENIX Security 16, Austin, TX, USA, August 10-12, 2016.*, 2016, pp. 279–296. [Online]. Available: https://www.usenix.org/conference/usenixsecurity16/technical-sessions/presentation/kogias

[5] L. Luu, V. Narayanan, C. Zheng, K. Baweja, S. Gilbert, and P. Saxena, "A secure sharding protocol for open blockchains," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, 2016, pp. 17–30. [Online]. Available: https://doi.org/10.1145/2976749.2978389

[6] V. Buterin and V. Griffith, "Casper the friendly finality gadget," *CoRR*, vol. abs/1710.09437, 2017. [Online]. Available: http://arxiv.org/abs/1710.09437

[7] M. Zamani, M. Movahedi, and M. Raykova, "Rapidchain: Scaling blockchain via full sharding," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, 2018, pp. 931–948. [Online]. Available: https://doi.org/10.1145/3243734.3243853

[8] E. Kokoris-Kogias, P. Jovanovic, L. Gasser, N. Gailly, E. Syta, and B. Ford, "Omniledger: A secure, scale-out, decentralized ledger via sharding," in *2018 IEEE Symposium on Security and Privacy, SP 2018, Proceedings, 21-23 May 2018, San Francisco, California, USA*, 2018, pp. 583–598. [Online]. Available: https://doi.org/10.1109/SP.2018.000-5

[9] J. Wang and H. Wang, "Monoxide: Scale out blockchains with asynchronous consensus zones," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*. Boston, MA: USENIX Association, 2019, pp. 95–112. [Online]. Available: https://www.usenix.org/conference/nsdi19/presentation/wang-jiaping

[10] M. Al-Bassam, A. Sonnino, S. Bano, D. Hrycyszyn, and G. Danezis, "Chainspace: A sharded smart contracts platform," in *25th Annual Network and Distributed System Security Symposium, NDSS 2018, San Diego, California, USA, February 18-21, 2018*, 2018. [Online]. Available: http://wp.internetsociety.org/ndss/wp-content/uploads/sites/25/2018/02/ndss2018_09-2_Al-Bassam_paper.pdf

[11] H. A. Kalodner, S. Goldfeder, X. Chen, S. M. Weinberg, and E. W. Felten, "Arbitrum: Scalable, private smart contracts," in *27th USENIX Security Symposium, USENIX Security 2018, Baltimore, MD, USA, August 15-17, 2018.*, 2018, pp. 1353–1370. [Online]. Available: https://www.usenix.org/conference/usenixsecurity18/presentation/kalodner

[12] S. Dziembowski, S. Faust, and K. Hostáková, "General state channel networks," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, CCS 2018, Toronto, ON, Canada, October 15-19, 2018*, 2018, pp. 949–966. [Online]. Available: https://doi.org/10.1145/3243734.3243856

[13] J. Teustch and C. Reitwießner, "A scalable verification solution for blockchains," 03 2017. [Online]. Available: https://truebit.io/

[14] S. Jain, P. Saxena, F. Stephan, and J. Teutsch, "How to verify computation with a rational network," *CoRR*, vol. abs/1606.05917, 2016. [Online]. Available: http://arxiv.org/abs/1606.05917

[15] C. Dong, Y. Wang, A. Aldweesh, P. McCorry, and A. van Moorsel, "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, CCS 2017, Dallas, TX, USA, October 30 - November 03, 2017*, 2017, pp. 211–227. [Online]. Available: https://doi.org/10.1145/3133956.3134032

[16] S. Das, V. J. Ribeiro, and A. Anand, "YODA: enabling computationally intensive contracts on blockchains with byzantine and selfish nodes," *CoRR*, vol. abs/1811.03265, 2018. [Online]. Available: http://arxiv.org/abs/1811.03265

[17] A. Kothapalli, A. Miller, and N. Borisov, "Smartcast: An incentive compatible consensus protocol using smart contracts," in *International Conference on Financial Cryptography and Data Security*. Springer, 2017, pp. 536–552.

[18] E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Succinct non-interactive zero knowledge for a von neumann architecture," in *23rd USENIX Security Symposium USENIX Security 14*, 2014, pp. 781–796.

[19] R. S. Wahby, I. Tzialla, A. Shelat, J. Thaler, and M. Walfish, "Doubly-efficient zksnarks without trusted setup," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 926–943.

[20] R. Kumaresan and I. Bentov, "Amortizing secure computation with penalties," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '16. New York, NY, USA: ACM, 2016, pp. 418–429. [Online]. Available: http://doi.acm.org/10.1145/2976749.2978424

[21] I. Bentov and R. Kumaresan, "How to use bitcoin to design fair protocols," in *International Cryptology Conference*. Springer, 2014, pp. 421–439.

[22] A. S. Aiyer, L. Alvisi, A. Clement, M. Dahlin, J. Martin, and C. Porth, "BAR fault tolerance for cooperative services," in *Proceedings of the 20th ACM Symposium on Operating Systems Principles 2005, SOSP 2005, Brighton, UK, October 23-26, 2005*, 2005, pp. 45–58. [Online]. Available: https://doi.org/10.1145/1095810.1095816

[23] G. Asharov, R. Canetti, and C. Hazay, "Towards a game theoretic view of secure computation," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*. Springer, 2011, pp. 426–445.

[24] R. Pass and J. Halpern, "Game theory with costly computation: formulation and application to protocol security," in *Proceedings of the behavioral and quantitative game theory: conference on future directions*. ACM, 2010, p. 89.

[25] I. Abraham, L. Alvisi, and J. Y. Halpern, "Distributed computing meets game theory: combining insights from two fields," *Acm Sigact News*, vol. 42, no. 2, pp. 69–76, 2011.

[26] N. Nisan, T. Roughgarden, E. Tardos, and V. V. Vazirani, *Algorithmic Game Theory*. New York, NY, USA: Cambridge University Press, 2007.

[27] C. Feige, K.-M. Ehrhart, and J. Krämer, "Voting on contributions to a threshold public goods game: An experimental investigation," Karlsruhe Institute of Technology (KIT), Department of Economics and Business Engineering, Working Paper Series in Economics 60, 2014. [Online]. Available: https://ideas.repec.org/p/zbw/kitwps/60.html

[28] J. Wang, F. Fu, T. Wu, and L. Wang, "Emergence of social cooperation in threshold public goods games with collective risk," *Physical Review E*, vol. 80, no. 1, Jul. 2009. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.80.016101

[29] E. Xiao and H. Kunreuther, "Punishment and cooperation in stochastic social dilemmas," *Journal of Conflict Resolution*, vol. 60, no. 4, pp. 670–693, 2016.

[30] L. Luu, J. Teutsch, R. Kulkarni, and P. Saxena, "Demystifying incentives in the consensus computer," in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, Denver, CO, USA, October 12-16, 2015*, 2015, pp. 706–719. [Online]. Available: https://doi.org/10.1145/2810103.2813659

[31] I. Abraham, D. Malkhi, K. Nayak, L. Ren, and A. Spiegelman, "Solidus: An incentive-compatible cryptocurrency based on permissionless byzantine consensus," *CoRR*, vol. abs/1612.02916, 2016. [Online]. Available: http://arxiv.org/abs/1612.02916

[32] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," 2009. [Online]. Available: http://www.bitcoin.org/bitcoin.pdf

[33] G. Wood, "Ethereum: A secure decentralised generalised transaction ledger eip-150 revision (759dccd - 2017-08-07)," 2017, accessed: 2018-01-03. [Online]. Available: https://ethereum.github.io/yellowpaper/paper.pdf

[34] I. Bentov, C. Lee, A. Mizrahi, and M. Rosenfeld, "Proof of activity: Extending bitcoin's proof of work via proof of stake," *SIGMETRICS Performance Evaluation Review*, vol. 42, no. 3, pp. 34–37, 2014. [Online]. Available: https://doi.org/10.1145/2695533.2695545

[35] B. Ford, "Untangling Mining Incentives in Bitcoin and ByzCoin," 2016. [Online]. Available: https://bford.github.io/2016/10/25/mining/

[36] SECBIT, "How the winner got Fomo3d prize — A Detailed Explanation," Aug. 2018. [Online]. Available: https://medium.com/coinmonks/how-the-winner-got-fomo3d-prize-a-detailed-explanation-b30a69\b7813f

[37] I. Eyal, A. E. Gencer, E. G. Sirer, and R. van Renesse, "Bitcoin-ng: A scalable blockchain protocol," in *13th USENIX Symposium on Networked Systems Design and Implementation, NSDI 2016, Santa Clara, CA, USA, March 16-18, 2016*, 2016, pp. 45–59. [Online]. Available: https://www.usenix.org/conference/nsdi16/technical-sessions/presentation/eyal

[38] R. Pass and E. Shi, "Fruitchains: A fair blockchain," in *Proceedings of the ACM Symposium on Principles of Distributed Computing*, ser. PODC 2017. New York, NY, USA: ACM, 2017, pp. 315–324. [Online]. Available: http://doi.acm.org/10.1145/3087801.3087809

[39] S. Azouvi, P. McCorry, and S. Meiklejohn, "Betting on blockchain consensus with fantomette," *CoRR*, vol. abs/1805.06786, 2018. [Online]. Available: http://arxiv.org/abs/1805.06786

[40] S. Bano, A. Sonnino, M. Al-Bassam, S. Azouvi, P. McCorry, S. Meiklejohn, and G. Danezis, "Consensus in the age of blockchains," *CoRR*, vol. abs/1711.03936, 2017. [Online]. Available: http://arxiv.org/abs/1711.03936

[41] M. Castro and B. Liskov, "Practical byzantine fault tolerance," in *Proceedings of the Third USENIX Symposium on Operating Systems Design and Implementation (OSDI), New Orleans, Louisiana, USA,*

February 22-25, 1999, 1999, pp. 173–186. [Online]. Available: https://dl.acm.org/citation.cfm?id=296824

[42] V. King and J. Saia, "Breaking the o (n^2) bit barrier: scalable byzantine agreement with an adaptive adversary," *Journal of the ACM (JACM)*, vol. 58, no. 4, p. 18, 2011.

[43] L. Ren, K. Nayak, I. Abraham, and S. Devadas, "Practical synchronous byzantine consensus," *arXiv preprint arXiv:1704.02397*, 2017.

[44] A. Miller, Y. Xia, K. Croman, E. Shi, and D. Song, "The honey badger of BFT protocols," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna, Austria, October 24-28, 2016*, 2016, pp. 31–42. [Online]. Available: https://doi.org/10.1145/2976749.2978399

[45] Z. Ren, K. Cong, T. Aerts, B. de Jonge, A. Morais, and Z. Erkin, "A scale-out blockchain for value transfer with spontaneous sharding," in *Crypto Valley Conference on Blockchain Technology, CVCBT 2018, Zug, Switzerland, June 20-22, 2018*, 2018, pp. 1–10. [Online]. Available: https://doi.org/10.1109/CVCBT.2018.00006

[46] M. H. Manshaei, M. Jadliwala, A. Maiti, and M. Fooladgar, "A game-theoretic analysis of shard-based permissionless blockchains," *IEEE Access*, vol. 6, pp. 78 100–78 112, 2018. [Online]. Available: https://doi.org/10.1109/ACCESS.2018.2884764

[47] C. Ho, R. van Renesse, M. Bickford, and D. Dolev, "Nysiad: Practical protocol transformation to tolerate byzantine failures," in *5th USENIX Symposium on Networked Systems Design & Implementation, NSDI 2008, April 16-18, 2008, San Francisco, CA, USA, Proceedings*, 2008, pp. 175–188. [Online]. Available: http://www.usenix.org/events/nsdi08/tech/full_papers/ho/ho.pdf

[48] A. Haeberlen, P. Kouznetsov, and P. Druschel, "The case for byzantine fault detection." in *HotDep*, 2006.

[49] A. Clement, H. Li, J. Napper, J.-P. Martin, L. Alvisi, and M. Dahlin, "Bar primer," in *2008 IEEE International Conference on Dependable Systems and Networks With FTCS and DCC (DSN)*. IEEE, 2008, pp. 287–296.

[50] M. Archetti and I. Scheuring, "Review: Game theory of public goods in one-shot social dilemmas without assortment," *Journal of Theoretical Biology*, vol. 299, pp. 9–20, 2012, exported from https://app.dimensions.ai on 2019/03/04. [Online]. Available: https://app.dimensions.ai/details/publication/pub.1014302925

[51] R. M. Dawes, "Social dilemmas," *Annual review of psychology*, vol. 31, no. 1, pp. 169–193, 1980.

[52] A. Groce, J. Katz, A. Thiruvengadam, and V. Zikas, "Byzantine agreement with a rational adversary," in *Automata, Languages, and Programming - 39th International Colloquium, ICALP 2012, Warwick, UK, July 9-13, 2012, Proceedings, Part II*, 2012, pp. 561–572. [Online]. Available: https://doi.org/10.1007/978-3-642-31585-5_50

[53] J. Eberhardt and S. Tai, "On or off the blockchain? insights on off-chaining computation and data," in *Service-Oriented and Cloud Computing - 6th IFIP WG 2.14 European Conference, ESOCC 2017, Oslo, Norway, September 27-29, 2017, Proceedings*, 2017, pp. 3–15. [Online]. Available: https://doi.org/10.1007/978-3-319-67262-5_1

[54] P. McCorry, S. Bakshi, I. Bentov, A. Miller, and S. Meiklejohn, "Pisa: Arbitration outsourcing for state channels," *IACR Cryptology ePrint Archive*, vol. 2018, p. 582, 2018. [Online]. Available: https://eprint.iacr.org/2018/582

[55] J. C. Harsanyi, "A new theory of equilibrium selection for games with complete information," *Games and Economic Behavior*, vol. 8, no. 1, pp. 91–122, 1995.

[56] T. Sasaki and S. Uchida, "Rewards and the evolution of cooperation in public good games," *Biology letters*, vol. 10, no. 1, p. 20130903, 2014.

[57] J. Sonnemans, A. Schram, and T. Offerman, "Public good provision and public bad prevention: The effect of framing," *Journal of Economic Behavior & Organization*, vol. 34, no. 1, pp. 143–161, 1998.

[58] F. Bolle and J. Spiller, "Not efficient but payoff dominant: Experimental investigations of equilibrium play in binary threshold public good games," Discussion Paper, Tech. Rep., 2016.

[59] A. Ertan, T. Page, and L. Putterman, "Who to punish? individual decisions and majority rule in mitigating the free rider problem," *European Economic Review*, vol. 53, no. 5, pp. 495–511, 2009.

[60] M. Andrychowicz and S. Dziembowski, "Pow-based distributed cryptography with no trusted setup," in *Annual Cryptology Conference*. Springer, 2015, pp. 379–399.

[61] C. Badertscher, J. A. Garay, U. Maurer, D. Tschudi, and V. Zikas, "But why does it work? A rational protocol design treatment of bitcoin," in *Advances in Cryptology - EUROCRYPT 2018 - 37th Annual International Conference on the Theory and Applications of Cryptographic Techniques, Tel Aviv, Israel, April 29 - May 3, 2018 Proceedings, Part II*, 2018, pp. 34–65. [Online]. Available: https://doi.org/10.1007/978-3-319-78375-8_2

[62] R. Ameer, "What is ethereum gas: Step-by-step guide," 2018. [Online]. Available: https://blockgeeks.com/guides/ethereum-gas-step-by-step-guide/

[63] E. Lombrozo, J. Lau, and P. Wuille, "Segregated witness (consensus layer)," 2018. [Online]. Available: https://github.com/bitcoin/bips/blob/master/bip-0141.mediawiki

[64] E. Syta, P. Jovanovic, E. Kokoris-Kogias, N. Gailly, L. Gasser, I. Khoffi, M. J. Fischer, and B. Ford, "Scalable bias-resistant distributed randomness," in *2017 IEEE Symposium on Security and Privacy, SP 2017, San Jose, CA, USA, May 22-26, 2017*, 2017, pp. 444–460. [Online]. Available: https://doi.org/10.1109/SP.2017.45

[65] D. Boneh, J. Bonneau, B. Bünz, and B. Fisch, "Verifiable delay functions," in *Advances in Cryptology - CRYPTO 2018 - 38th Annual International Cryptology Conference, Santa Barbara, CA, USA, August 19-23, 2018, Proceedings, Part I*, 2018, pp. 757–788. [Online]. Available: https://doi.org/10.1007/978-3-319-96884-1_25

[66] P. Feldman, "A practical scheme for non-interactive verifiable secret sharing," in *28th Annual Symposium on Foundations of Computer Science (sfcs 1987)*. IEEE, 1987, pp. 427–438.

[67] J. Kwon, "Tendermint: Consensus without mining," *Retrieved May*, vol. 18, p. 2017, 2014.

[68] B. C. Arnold, *Pareto distributions*. Chapman and Hall/CRC, 2015.

[69] J. Yu, D. Kozhaya, J. Decouchant, and P. Verissimo, "Repucoin: Your reputation is your power," *IEEE Transactions on Computers*, 2019.

[70] G. Wood, "Polkadot: Vision for a heterogeneous multi-chain framework," *White Paper*, 2016.

Appendix A
Faineant's Revelry

We present a profitable but selfish strategy which is feasible in most existing scalable frameworks, and reveal its potential sabotage to the validity of system:

At the Prepare phase of BFT agreement, each faineant sets up a freeload bound B_f and monitors the multicast channel, waiting for extra B_f valid prepared messages. When condition is satisfied, they multicast their own prepared messages and assume the leader's proposal is valid. Though taking no workload of validation throughout the game, faineants are indistinguishable from the advantaged participants (No matter they are altruistic or malicious).

We simulate the possible consequences triggered by the strategy. Consider the most common configuration in Byzantine agreement, a committee of n nodes including at most $f = \lfloor n/3 \rfloor$ Byzantine players manages to reach consensus on the validity of a transaction. The costly workload consumes C for each node's execution and awards R to the whole committee. If consensus is finally reached in some round[23], the reward will be shared among nodes who make positive responses.

Altruists always faithfully follows the protocol and commits the correct response, while Byzantine nodes collude to commit the wrong verdict rather than abort. To accurately follow the advantaged party and accelerate the consensus, Faineants initially set freeloading bound B_f as $\lfloor 2/3 \cdot n \rfloor$, and adjust it by $B_f = 1/2 \cdot B_f$ whenever there is a view change.

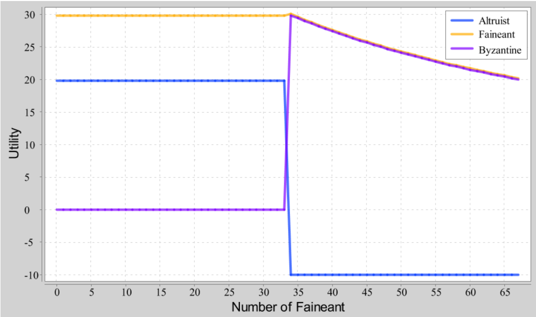


Fig. 19: Utilities of all roles vary from the number of Faineant($n = 100,\ R = 2000,\ C = 10,\ f = \lfloor n/3 \rfloor$).

Figure 19 shows the variation of utilities in the worst-case setting. Obviously, such freeloading trick is always the dominant strategy. Profits-driven faineants cooperate no matter when altruists or Byzantines take the dominance, taking the best reward as well as saving the expenses of validation. The situation is much more practical than the predicament of social dilemma where everybody aborts.

It is noteworthy that Byzantine nodes are more inflammatory to faineants as they deliberately make fast and consistent verdicts without execution, which leads to an outburst when the number of faineants just exceeds $\lceil 2/3 \cdot n \rceil - f$. Under the existing Byzantine consensus, this also marks the breaking of validity as the inevitable outcome due to rationality. Consider an extreme but perfectly possible case where validators are all rational, then a single malicious node could control the whole committee, making the protocol entirely lose its validity.

Appendix B
Imbalance of intact competition without work assignment

Employing intact competition on-chain ostensibly respects validators' free will and may lead to a faster validation speed. However, incompatible incentive would gradually make the system lose its balance. For instance, if validators can make Verify proposals in a spontaneous manner, workload repetition is evitable. As the reward is limited and deterministic, someone should lose his power and obtain nothing. Once the source of VIT is scarce, fierce competition could give rise to a negative expectation for rational validators, this in return makes abort as the best strategy under the BR assumption, thus VITs may lose their liveness.

In the long term, as it is unable to tune the frequency of incoming VITs under the open ecology, unhealthy competition could lead to a vicious circle. Unstable income could make the position of validator worthless to external candidates. The performance of system is additionally degraded, derives poor finality and robustness and finally results in the collapse of system. Even worse, freeloading may revive in this pattern. As a result, heavy cryptographic measures should be equipped, making the workflow much more complicate.

Appendix C
Research prospects

Our work breaks the bottleneck of on-chain validation, meanwhile provide useful insights on the design of incentive mechanism, Byzantine faults inhibition and ecological management in permissionless environment. During the study, we have discovered some interesting and non-trivial prospects based on our research, which deserve further explorations to step up the development of blockchain techniques and applications. Below, we share these topics, give brief discussions and look forward to subsequent cooperation and breakthroughs:

The Extreme of Game Theoretic Model. Since intensive workload is thoroughly decoupling from daily consensus, an equal distribution seems adequate and much safer to allocate basic rewards collecting from Byzantine consensus, which collectively compose a plausible global incentive mechanism with Lever. However, before making the assertion, there are still some challenges to address:

- If taken coalition among rational players into consideration, to prevent negative effects from mining pools, the model further needs to be analyzed and refined by insights on cooperative game theory.
- Some of the pivotal parameters (like k_1, k_2, W) could be dynamic in long term according to the balance of internal

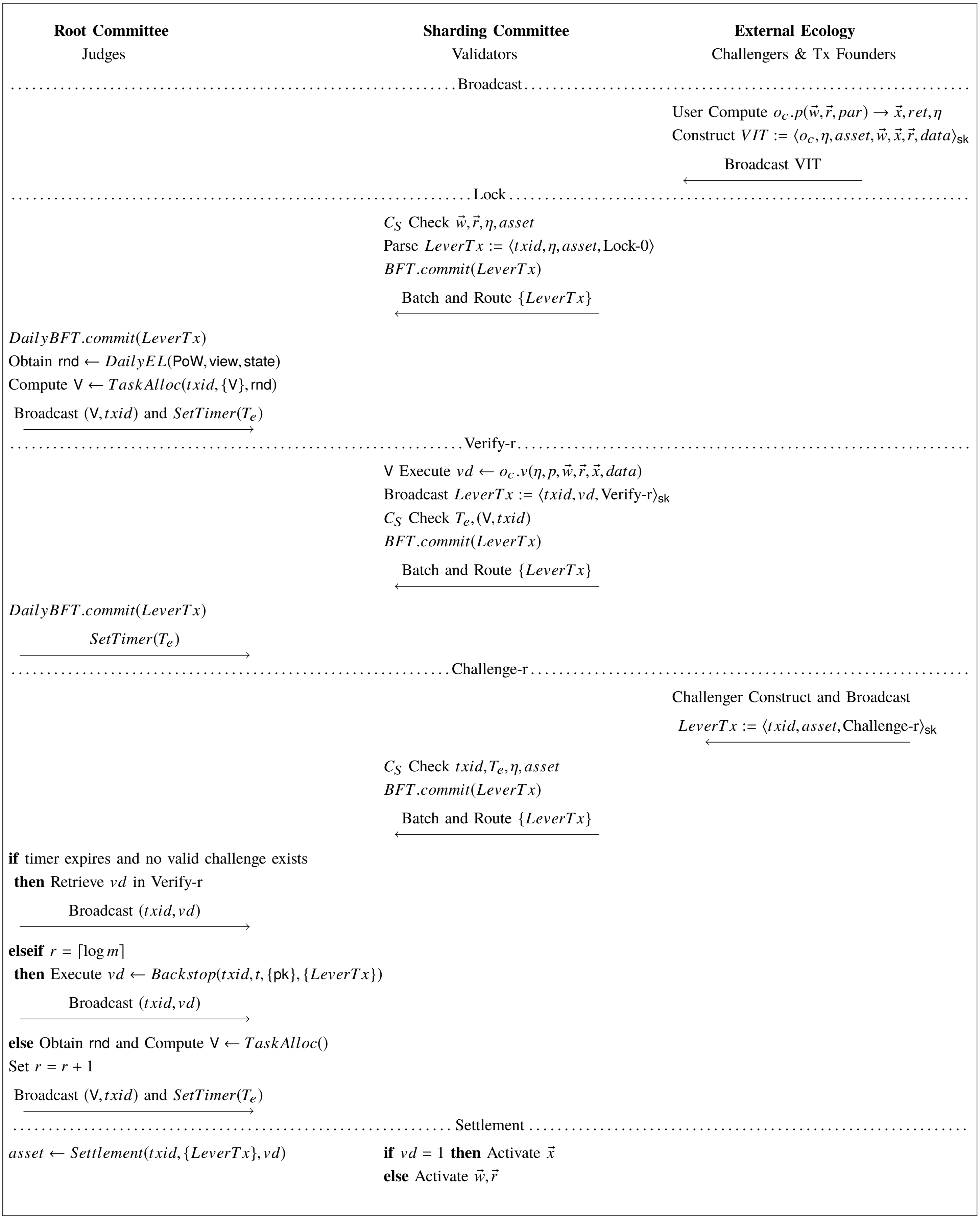
[23] We set the time as at least $\lfloor 2n/3 \rfloor + 1$ valid signatures w.r.t. one proposition are collected.

workload and external financial factors. There need fine-grained mechanisms to set up healthy feedback relations and lead the adjustments safely.

- In Finality game, if a container is unfortunately suspended, the protocol is delayed for m periods to ensure least intersection of validators in repeated games. It is quite ambitious to design protocols erasing such latency and explore the possible equilibriums in a re-voting protocol. An intuition would be cooperators in last game could become altruists of the next round. The hinge of design turns to motivate potential defectors to alter their votes while maintaining a fair allocation on incentive.

Peculiar Election via Challenge Interface. While most existing protocols rely on spontaneous behaviors of members to isolate discovered Byzantine nodes, we innovatively adopt an peculiar reconfiguration for reliable challengers to substitute the incumbent malicious validators. It therefore opens a specialized election channel for external rational candidates to join in the committee, meanwhile purifies the whole ecology. As C_R timely updates the membership according to the finality results, the consistency of mechanism is guaranteed. With more fine-grained procedure and detailed analyses, we believe it is possible to propose a general and effective Byzantine inhibition schema following such pattern.

Designs for Advanced Architectures We plan to build systems with greater ambitions on top of Lever. One is a fair reputation system applied for intensive validation, which could integrate more deterministic and comprehensive factors into the scope of endorsement, thus releasing the solid requirement on deposit and additionally expand the system up. The other is a validation framework supportive to heterogeneous requests from different autonomous systems. More challenges would be appeared in resolving inconsistent elements like token transformation, authority management, discrepant data format and committee configuration. Considering the prior attempts [69], [70], there is still plenty of expectations on such topics. We believe our framework could provide greater potentials.



Root Committee
Judges

Sharding Committee
Validators

External Ecology
Challengers & Tx Founders

Broadcast

User Compute $o_c.p(\vec{w},\vec{r},par) \rightarrow \vec{x}, ret, \eta$
Construct $VIT := \langle o_c, \eta, asset, \vec{w}, \vec{x}, \vec{r}, data\rangle_{\mathsf{sk}}$
$\xleftarrow{\text{Broadcast VIT}}$

Lock

C_S Check $\vec{w},\vec{r},\eta, asset$
Parse $LeverTx := \langle txid, \eta, asset, \text{Lock-0}\rangle$
$BFT.commit(LeverTx)$
$\xleftarrow{\text{Batch and Route } \{LeverTx\}}$

$DailyBFT.commit(LeverTx)$
Obtain $\mathsf{rnd} \leftarrow DailyEL(\mathsf{PoW}, \mathsf{view}, \mathsf{state})$
Compute $\mathsf{V} \leftarrow TaskAlloc(txid, \{\mathsf{V}\}, \mathsf{rnd})$
$\xrightarrow{\text{Broadcast } (\mathsf{V}, txid) \text{ and } SetTimer(T_e)}$

Verify-r

V Execute $vd \leftarrow o_c.v(\eta, p, \vec{w}, \vec{r}, \vec{x}, data)$
Broadcast $LeverTx := \langle txid, vd, \text{Verify-r}\rangle_{\mathsf{sk}}$
C_S Check $T_e, (\mathsf{V}, txid)$
$BFT.commit(LeverTx)$
$\xleftarrow{\text{Batch and Route } \{LeverTx\}}$

$DailyBFT.commit(LeverTx)$
$\xrightarrow{SetTimer(T_e)}$

Challenge-r

Challenger Construct and Broadcast
$\xleftarrow{LeverTx := \langle txid, asset, \text{Challenge-r}\rangle_{\mathsf{sk}}}$

C_S Check $txid, T_e, \eta, asset$
$BFT.commit(LeverTx)$
$\xleftarrow{\text{Batch and Route } \{LeverTx\}}$

```
if timer expires and no valid challenge exists
  then Retrieve $vd$ in Verify-r
    Broadcast $(txid, vd)$ →
elseif $r = \lceil \log m \rceil$
  then Execute $vd \leftarrow Backstop(txid, t, \{\mathsf{pk}\}, \{LeverTx\})$
    Broadcast $(txid, vd)$ →
else Obtain rnd and Compute $\mathsf{V} \leftarrow TaskAlloc()$
Set $r = r + 1$
  Broadcast $(\mathsf{V}, txid)$ and $SetTimer(T_e)$ →
```

Settlement

$asset \leftarrow Settlement(txid, \{LeverTx\}, vd)$

if $vd = 1$ **then** Activate $\vec{x}$
else Activate $\vec{w},\vec{r}$

Fig. 20: Integrated Protocol of Lever