Facilitating Serverless Match-based Online Games with Novel Blockchain Technologies

10

FEIJIE WU, Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR, China
HO YIN YUEN, Department of Biomedical Engineering, The Hong Kong Polytechnic University, Hong Kong SAR, China
HENRY CHAN, Department of Computing, The Hong Kong Polytechnic University, Hong Kong SAR, China
VICTOR C. M. LEUNG, College of Computer Science and Software Engineering, Shenzhen University, China
and Department of Electrical and Computer Engineering, The University of British Columbia, Canada
WEI CAI, School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, China and Shenzhen Institute of Artificial Intelligence and Robotics for Society, China

Applying **peer-to-peer (P2P)** architecture to online video games has already attracted both academic and industrial interests, since it removes the need for expensive server maintenance. However, there are two major issues preventing the use of a P2P architecture, namely how to provide an effective distributed data storage solution, and how to tackle potential cheating behaviors. Inspired by emerging blockchain techniques, we propose a novel consensus model called **Proof-of-Play (PoP)** to provide a decentralized data storage system that incorporates an anti-cheating mechanism for P2P games, by rewarding players that interact with the game as intended, along with consideration of security measures to address the Nothing-at-stake Problem and the Long-range Attack. To validate our design, we utilize a game-theory model to show that under certain assumptions, the integrity of the PoP system would not be undermined due to the best interests of any user. Then, as a proof-of-concept, we developed a P2P game (*Infinity Battle*) to demonstrate how a game can be integrated with PoP in practice. Finally, experiments were conducted to study PoP in comparison with **Proof-of-Work (PoW)** to show its advantages in various aspects.

This work was supported by Shenzhen Science and Technology Program (Grant No. JCYJ2021032412 4205016), by the Shenzhen Institute of Artificial Intelligence and Robotics for Society (AIRS), by the Department of Computing, The Hong Kong Polytechnic University (account number ZVQ5), by Guangdong Pearl River Talent Recruitment Program (Grant No. 2019ZT08X603), and by Shenzhen Science and Technology Innovation Commission (Grant No. R2020A045).
Authors' addresses: F. Wu and H. Chan, Department of Computing, The Hong Kong Polytechnic University, 11 Yuk Choi Road, Hung Hom, Kowloon, Hong Kong SAR, China; emails: harli.wu@connect.polyu.hk, cshchan@comp.polyu.edu.hk; H. Y. Yuen, Department of Biomedical Engineering, The Hong Kong Polytechnic University, 11 Yuk Choi Road, Hung Hom, Kowloon, Hong Kong SAR, China; email: andy.aa.yuen@connect.polyu.hk; V. C. M. Leung, College of Computer Science and Software Engineering, Shenzhen University, 3688 Nanhai Ave., Shenzhen 518060, Guangdong, China and Department of Electrical and Computer Engineering, The University of British Columbia, Vancouver, 2332 Main Mall, British Columbia, Canada V6T 1Z4; email: vleung@ieee.org; W. Cai (corresponding author), School of Science and Engineering, The Chinese University of Hong Kong, Shenzhen, 2001 Longxiang Ave, Longgang District, Shenzhen 518172, Guangdong, China and Shenzhen Institute of Artificial Intelligence and Robotics for Society, 14F, Tower G2, Xinghe World, Rd Yabao, Longgang District, Shenzhen 518129, Guangdong, China; email: caiwei@cuhk.edu.cn.

1533-5399/2023/02-ART10 $15.00
https://doi.org/10.1145/3565884

CCS Concepts: • **Computer systems organization** → **Peer-to-peer architectures**; • **Applied computing** → *Computer games*; • **Information systems** → Distributed storage;

Additional Key Words and Phrases: Peer-to-peer game, blockchain, consensus model

ACM Reference format:
Feijie Wu, Ho Yin Yuen, Henry Chan, Victor C. M. Leung, and Wei Cai. 2023. Facilitating Serverless Match-based Online Games with Novel Blockchain Technologies. *ACM Trans. Internet Technol.* 23, 1, Article 10 (February 2023), 26 pages.
https://doi.org/10.1145/3565884

1 INTRODUCTION

With the rapid development of electronic sports, live streaming, and the recent call for support of the metaverse concept [18], video games are embracing unprecedented prosperity. One popular type of game is match-based games in which two groups of players compete with one another for the final honor in every match. In particular, **Multiplayer Online Battle Arena (MOBA)** is a subset of match-based games, which have attracted considerable interest in recent years. Traditional match-based online games, such as League of Legends[1] typically rely on a group of dedicated gaming servers. These client-server models are a centralized approach that unfortunately comes with a number of disadvantages, such as single-point-of-failure, possibly high maintenance costs for the dedicated servers, and inflexible operations.

Peer-to-peer (P2P) architecture provides an ideal solution to address the aforementioned issues due to its decentralized nature, and numerous works are actively supporting such an approach [4, 5, 14, 48, 67]. Decentralizing game servers would not only eliminate a large portion of the maintenance cost for developers, but would also allow game servers to scale automatically according to the number of players. The nature of decentralization also enables enthusiasts to sustain the game ecosystem for many years to come. As a result, the multiplayer system of many games, such as the Monster Hunter series,[2] enjoys long-lasting popularity owing to its use of the P2P multiplayer solution.

Nevertheless, P2P solutions to-date never made a full-scale impact in the video gaming market. Some famous games now provide an option for running on P2P networks, but the gaming data can neither be authenticated nor permanently stored outside of an exclusive network [66]. Moreover, without a centralized authority, insincere users will try to tamper with their game data, which is usually stored locally, to achieve their interest, i.e., cheating. One of the examples that suffers from this problem would be the famous Diablo II: Resurrected.[3] In Diablo II, part of its infrastructure relies on a P2P system, and so cheating clients can alter the in-game objects by manipulating their local data to defeat honest players in an unfair way. This discourages honest players from investing their time and money in the game, and therefore later on, Blizzard removed the peer-to-peer support for Diablo II, so that the game data is instead processed in a centralized but more secured manner (i.e., official dedicated servers) to prevent these potential security risks.

As an emerging technology, blockchain [41] provides a decentralized, transparent, and trustless data storage solution [39, 73]. In this case, a blockchain can be used as a decentralized database for storing gaming data for P2P games. This can ensure that nobody can interpolate or modify existing data, making blockchain a reliable and effective solution for the aforementioned problems. Generally, blockchain systems are categorized into three types in terms of accessibility: public blockchain,

[1] https://www.leagueoflegends.com/.
[2] https://www.monsterhunterworld.com/.
[3] https://www.diabloii.net/.

permissioned blockchain, and private blockchain. The latter two types assume the explicit existence of third parties to validate all users and transactions, weakening the credential of virtual items in an online game. As for public blockchain systems, consensus algorithms can be broadly categorized into two types: proof-based and voting-based. For proof-based models [2, 19, 38, 49], since every block requires verification by distributed methods (e.g., zero-knowledge proof) [44], they typically require a commitment of computational units. An example would be the famous "Bitcoin mining", which requires committing CPU powers to host the blockchain. Layer-2,[4] albeit cost-efficient to proceed with transactions, requires an endorsement/assistance from a third party, which impairs the decentralized feature of a blockchain. Besides, as the most common practice on Layer-2 blockchains, cross-chain applications are highly controversial, especially in terms of security performance.[5] The recent Ronin Bridge Attack[6] has attracted the public concerns and doubts on the feasibility of a Layer-2 blockchain. For vote-based models [29, 30, 64], e.g., **Proof-of-Stake (PoS)**, they use stake-based algorithms to maintain a blockchain via a voting-related mechanism. If either model is applied to a P2P game, it would be unintuitive and require a commitment beyond playing the game, effectively barricading the majority of the gamer population. Specifically, since the direct implication of the nothing-at-stake problem would be that it is more likely to have forks in a blockchain (conflicts in data consensus), this increases the likelihood where game data of a player fail to be finalized on a blockchain. Furthermore, the economic model that maintains the stability of PoS may not be directly applicable for scenarios like video games, creating yet another overhead to implement blockchains in video games. To resolve this, we would need an application-specific blockchain that naturally integrates into the context of P2P games.

Considering that existing consensus models are not suitable for serverless match-based online games, we introduce a novel consensus model – **Proof-of-Play (PoP)** for this purpose. PoP serves as a middleware between P2P games and its exclusive blockchain, so that players play the P2P game off-chain and synchronize the game data to the chain when it ends. Our design achieves three goals as follows: (1) any game is theoretically supported since it is only a middleware from game to blockchain; (2) historical data stored on the blockchain will be genuine; and (3) the blockchain will operate for as long as the game will. We achieve these goals by having all players (i.e., PoP nodes, the blockchain users) to be miners, and setting it up so that playing the game is equivalent to mining a block. In PoP, the harder the player plays, the faster it mines. It is also designed to resist various attacks, such as Long-range Attack. Furthermore, none of the PoP nodes is required to keep running on a 24/7 basis, nor is there an incentive to. Instead, players run the blockchain because they play the games. All this combined consolidates a natural, secured, and decentralized gaming system.

PoP is constituted by three components – *Internal Negotiation*, *Decentralized Surveillance*, and *Block Mining* – and each contributes to an impeccable system. We summarize our contributions as follows:

- We present a novel PoP scheme for supporting P2P games. It consists of three innovative and effective processes: internal negotiation for reaching agreement among the players and protecting data integrity, decentralized surveillance for preventing illegal behaviours, and block mining for generating blocks for the blockchain.
- We formulate a game theory model to prove several theorems to support the effective operation of PoP under certain assumptions. In particular, we verify there is only one Nash equilibrium point such that all players should tell the truth to maximize their own interests.

[4] https://academy.binance.com/en/glossary/layer-2.
[5] https://old.reddit.com/r/ethereum/comments/rwojtk/ama_we_are_the_efs_research_team_pt_7_07_january/hrngyk8/.
[6] https://cryptopotato.com/the-biggest-ever-crypto-hack-what-happened-in-the-ronin-bridge-attack/.

- We present a game prototype (i.e., an open-source P2P game Infinity Battle [66][7]) based on PoP for proof-of-concept purposes. It verifies the concept and operation of PoP using a real game.
- We discuss experimental/simulation results to compare PoP and PoW, highlighting the advantages in various aspects.

This paper is a follow-up work to two previous conference papers, namely, (I) Proof-of-Play: A Novel Consensus Model for Blockchain-based Peer-to-Peer Gaming System [70],[8] and (II) Infinity Battle: A Glance at How Blockchain Techniques Serve in a Serverless Gaming System [66]. Other than the existing work, this paper includes comprehensive related work, discusses the PoP system in more detail, formulates a complete game theory model with theorems and provides more experimental and simulation results.

The rest of the paper is organized as follows: we first discuss the related works on the gaming system using blockchain in Section 2. Then, Section 3 presents the three components of the proposed Proof-of-Play consensus model, including *Internal Negotiation*, *Decentralized Surveillance*, and *Block Mining*. To validate the proposed model and to measure its efficiency, Section 4 provides numerical results for *Internal Negotiation* and *Block Mining*. Afterwards, the *Infinity Battle*, a P2P turn-based strategy game integrated with PoP, is introduced in Section 5. Finally, Section 6 summarizes the paper with conclusions and future research directions.

2 RELATED WORK

2.1 Peer-to-Peer Network and Online Games

The main feature of a P2P network is that each node is a resource provider as well as a resource requestor [23, 62]. In general, there are two categories of P2P networks – "Hybrid" and "Pure" [53]. In a pure peer-to-peer networking scenario, every node is equal and is a host of the network, whereas in a hybrid structure, there is a central committee to bootstrap the entire whole network. In general, peer-to-peer architecture is different than client-server architecture, in that all users maintain a backup version simultaneously, and they are always ready to bootstrap the network [57].

Over the years, P2P technologies have become more mature and have been applied in a wide range of areas, including the video game industry [3, 17, 21, 22, 24]. The idea [32] was initially proposed to enhance the scalability of **Massively Multiplayer Games (MMGs)**, with two major research problems – optimizing system performance, and securing data synchronization. Many researchers have attempted to tackle these two problems and design various models. For example, system performance can be optimized by reducing network latency [1, 50, 54, 71], and data integrity is secured with proper detection and prevention of malicious data modification i.e., cheating [27, 28, 37, 63].

2.2 Blockchain and Blockchain Games

Blockchain comprises of a fixed sequence of blocks, where information written on each block is immutable. Using a pure P2P network architecture, a blockchain [41] is held and agreed on by all members of the system. In practice, a blockchain is represented as a single linked list, with a consensus mechanism as a protocol to govern the addition of each new block. All blocks are validated and protected by means of cryptographic methods. While blockchain was first proposed for supporting cryptocurrencies (e.g., Bitcoin [41]), it has now emerged to support P2P smart contracts in

[7]Code available at GitHub: https://github.com/HCSLab/InfinityBattle.
[8]This conference paper received a Best Student Paper award from the 2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure (BSCI'19).

general. There has also been the development of blockchain-based ecosystems, such as Ethereum [7]. Today, blockchain-enabled **decentralized applications (dApps)** have been developed in a wide range of areas, such as the IoT [15, 25, 45, 55], which is creating a new paradigm [9].

A blockchain game refers to a P2P game that operates over a blockchain system. Interactions between the players and the blockchain manifest as the players play the game [40]. Some public platforms, such as EOS[9] [26] and Ethereum[10] [7], provide **blockchain as a service (BaaS)** [31, 46, 52] to reduce the cost of development in building a private chain (e.g., for blockchain-based games). In recent years, many blockchain games [10, 40, 61] have emerged. For example, CryptoKitties,[11] a web-based kitty collection game, utilizes Ethereum to store the historical details of the game, including a kitty's creation, breed, and trade. Players can give birth to a baby kitty by combining other kitties owned by the player, and sell kitties for virtual coins. So far, only turn-based or static web games can be found in the Ethereum library, due to the technical challenges involved in developing dynamic and real-time blockchain games. Blockchain games are promising, and more research is required.

2.3 Game-Related Consensus Models

A public blockchain system enjoys high flexibility that allows a user to join or leave the system freely [72]. Therefore, a consensus model is required to ensure that all participants agree on the same dataset in a distributed manner. For instance, under **Proof-of-Work (PoW)** [2], a well-known proof-based consensus algorithm, a block is verified only when its hash value is smaller than a threshold, which is dynamic and traceable on the mainchain.

Apart from general consensus models, there are also game-specific consensus models. These models are specifically designed for games, taking into consideration some special requirements. For example, BUFF[12] [58] introduces an optimized DPoS by electing 21 candidate nodes to generate the next block and reward successful miners. Naturally, decentralization is maintained since miners would not want to collude and hence undermine user trust in the blockchain. Huntercoin[13] adopts PoW and presents the concept of Human (or AI) mining: approximately 80% of new coins are obtainable by ordinary players while the rest are given to the miner as a reward; a transaction is conducted through a battle between two players.

2.4 Blockchain Security

Concerns over security on the blockchain continue to increase as it becomes increasingly popular, especially in the Fintech industry due to the explosion in popularity of Bitcoin [35, 36]. The following provides a summary of major threats that may weaken the reliability of existing blockchain consensus models.

Sybil Attack: An aggressor can initiate Sybil Attack [16] in a P2P architecture by creating numerous pseudonymous identities and using them to control the whole network, subverting trust in a blockchain. In a blockchain ecosystem, one of the most significant attacks is the 51% attack, which means that an attacker gains majority control of the distributed ledgers. PoW tackles this attack by ensuring that it is practically impossible for a node to control 51% of the total hashing power, and PoS alleviates it by reducing the incentive to attack due to its underlying economic theory [30].

[9] https://eos.io/.
[10] https://www.ethereum.org.
[11] https://www.cryptokitties.co/.
[12] https://buff.game.
[13] https://xaya.io/huntercoin-legacy/.

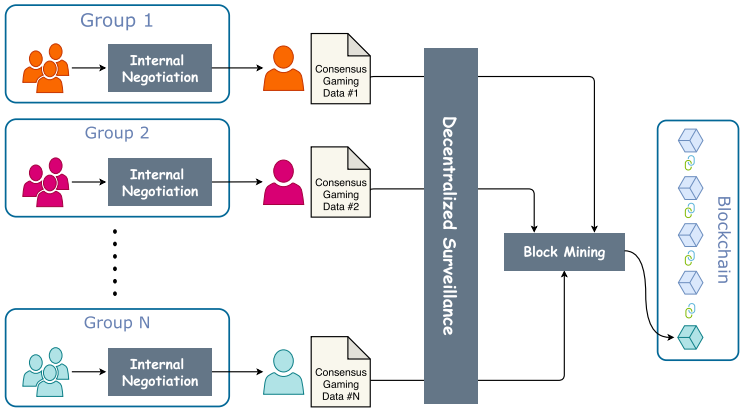


Fig. 1. Proof-of-play overview.

Nothing-at-stake Problem: Due to network latency, some nodes may not be aware of some new blocks, so different nodes are committed to different versions of a blockchain, creating a fork. Due to this issue, an honest node should adapt the first-come principle to determine which fork should belong to the main chain when all forks have the same number of blocks. However, in some consensus algorithms, a miner can mine in different forks of the blockchain without spreading its mining resources too thinly. As a result, rationalists follow all forks to assure themselves they are getting more benefits as they mine on more chains. This is known as the Nothing-at-stake problem [6], which hinders the stability and the uniqueness of the main chain.

Long-range Attack: In this attack, an adversary attempts to fork a chain from a random position and make it longer than the main chain so that the new one can take its place. This illegal behavior is known as a long-range attack [12], which may occur in consensus models that do not require computational power, such as PoS. To tackle potential attacks, Nxt coin [64] sets a checkpoint for the system, indicating that modification is not allowed on a subchain where all blocks have at least 720 confirmations.

3 PROOF-OF-PLAY MODEL

In this paper, we propose a novel Proof-of-Play (PoP) scheme to seamlessly integrate a P2P game with a blockchain system. Different from other consensus models, users and miners are integrated, which means that a player is a block writer as well. Our main idea is that to the player, and there is no difference between playing and mining. Compared to incentive-driven blockchain systems, such as HunterCoin, it keeps the games focused on their innate nature, which is to provide entertainment.

There are three system components to realize these features, which are highlighted in Figure 1:

(1) **Internal Negotiation** safeguards the integrity and consistency of the game data
(2) **Decentralized Surveillance** enhances the effect of playing effort so as to avoid illegal behaviors occurring on the chain
(3) **Block Mining** limits the quantity of valid blocks to control the writing process of the blockchain

To elaborate on Figure 1, assume there are N different game matches waiting to be synchronized on the blockchain. In each group, all players will reach an agreement on the final result once the match ends. Then, the **most valuable player (MVP)** of the match will broadcast the result globally and compete with others for the next block writer. Before generating the new block, potential

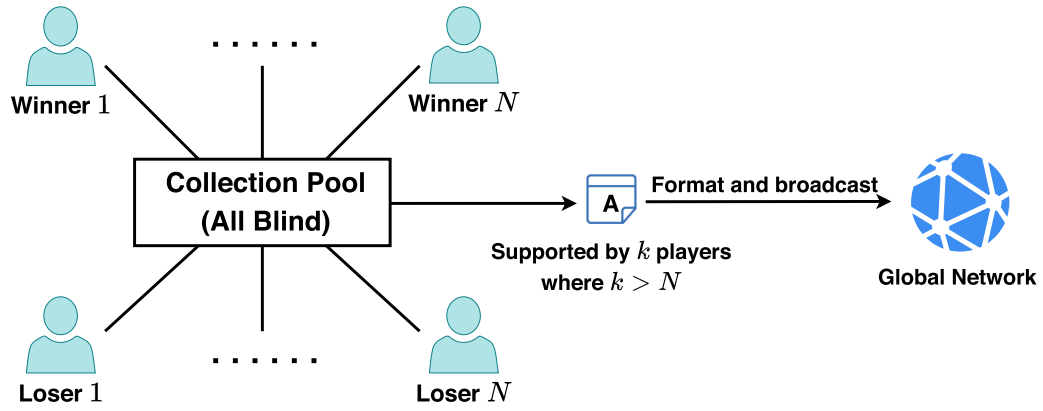


Fig. 2. Internal negotiation.

nominees are supposed to detect unauthorized actions and jot down relevant details as evidence. Afterward, in the mining session, mathematical methods are adopted to validate the statistics and filter the blocks. When a valid block appears, it is immediately written to the blockchain. Consequently, these N matches are included in the latest blockchain.

To explain the aforementioned system, the following three sections discuss each system element in detail. Last but not least, we identify the cheating issues in Section 3.4 and discuss how our proposed system defeats them case by case.

3.1 Internal Negotiation

As the first checkpoint in the system, Internal Negotiation seeks an agreement among all players as well as ensures the integrity of the game data. One of the first challenges here is to address malicious parties in the system that seek to sabotage and terminate the negotiation. Even more so, they may also forge match results that are beneficial for everyone to agree on, i.e., data that are the outcome of players' collusion instead of a record of the match, even if the cost of forging is expensive. So, to address a PoP blockchain being compromised in this way, we propose the following techniques to avoid the risk of conspiracy and reduce the probability of counterfeit, securing the idea of Internal Negotiation.

Assume that players in a match have no acquaintance with one another and therefore, a result supported by the majority is regarded as the true result. Kraft [33] comes up with an idea called Shared Turns on how two players disclose their moves simultaneously, in essence a commitment scheme. We expand on his idea to acquire a result acknowledged by a majority of participants, which is shown in Figure 2. Jointly maintained by all members, a collection pool is used to collect all possible results stated by each player, and its operations are given as follows:

Step 1: Obtain everyone's public key
Step 2: Get the hash value of match data and the digital signature from all players
Step 3: Validate each signature with the provided information
Step 4: Ask each player for the game results
Step 5: Verify each game result with the corresponding hash value

The partial/entire course of a match is included in the game result because it is a proof of play: the irreplaceable and dedicated playing efforts of the players. At the end of the collection pool, a result approved by the majority of players is available. The most valuable player (MVP) subsequently

wraps the result and broadcasts it globally. This negotiation step ensures that data will not be tampered with, since the signatures of all players must be presented for the data to be created.

Considering that a consensus may not be reached in some matches, a result determination mechanism is proposed in Section 3.1.1, encouraging everyone to be truthful in order to reach a consensus.

3.1.1 Trustworthiness in Internal Negotiation. Game theory makes use of mathematical models of strategic interactions among rational decision-makers in order to plan for best-case scenarios [47], with its application most widely known in economics as well as in various areas such as computing and algorithms. In a situation where a consensus cannot be reached for a match, we designed a game theory model to facilitate an agreement.

One of the key concerns of PoP is having a mechanism to address cheating behaviors in a match-based P2P game. Consider that there is only one true result, a failure in consensus (i.e., inconsistent match results during Internal Negotiation), which implies cheating behaviors. Definition 1 presents key notations throughout the section.

Definition 1 (Key Notation). Given a match-based game with n players in each team:

- Victory Team (VT) = $\{w_1, \ldots, w_n\}$, Defeated Team (DT) = $\{u_1, \ldots, u_n\}$. $TL = VT \cup DT$ is a set of players in a match.
- $\Sigma = \Sigma_{w_1} \times \cdots \times \Sigma_{w_n} \times \Sigma_{u_1} \times \cdots \times \Sigma_{u_n}$ is a set of strategy profiles, where Σ_k is a set of valid game results of Player k, including a truth $\sigma_k^{(t)}$ and a lie $\sigma_k^{(l)}$.
- $\sigma = \sigma_{w_1} \times \cdots \times \sigma_{w_n} \times \sigma_{u_1} \times \cdots \times \sigma_{u_n}$ represents one of the portfolios in Σ, where $\sigma \in \Sigma$, and $\sigma_i \in \Sigma_i$ is a decision made by Player i. Thus, $\sigma_{-k} = \sigma \setminus \{\sigma_k\}$ is defined as any strategy profile σ without player k's action.
- $\Pi = \{\pi_{w_1}, \ldots, \pi_{w_n}, \pi_{u_1}, \ldots, \pi_{u_n}\}$ is a set of payoff functions evaluated at a value of σ. It indicates a reward distribution scenario based on the list of results assembled by all players.

Prior to designing an appropriate game theory model, we first build up a revenue allocation mechanism for a P2P game. Inspired by a typical reward scheme implemented by match-based games (e.g., Honor of Kings), we design the following reward allocation mechanism, which depends on P and γ.

Definition 2 (Reward Allocation Mechanism). The revenue allocation mechanism is designed/defined with the following constraints:

- The total reward for a match is P. Therefore, for any revenue ratio α in a valid game, the corresponding revenue is αP, which is rounded up to a positive integer to facilitate the computation by the Generating Function [34] in Proposition 1.
- There exists a scalar $\gamma > 0$ such that
  - Its half value is the minimum income for all players in any valid game result;
  - Suppose there exist two different game results, in which the revenues of Player i are α_i and $\tilde{\alpha}_i$. If $\tilde{\alpha}_{MVP} \leq \alpha_{MVP} - \gamma$ holds, we have

$$\forall i \in VT, \tilde{\alpha}_i \leq \alpha_i - \gamma/2; \quad \forall i \in DT, \tilde{\alpha}_i \leq \alpha_i + \gamma \tag{1}$$

Obviously, the minimum income can be defined manually by setting γ. Additionally, γ can regulate the revenue allocation schemes. Apart from the mechanism designed above, we make the following assumption that the winners, losers and MVP can always be clearly identified in a match, even if the final results vary among the players.

Table 1. Winners' and Losers' Revenue (i.e., payoff matrix)

VT Result	DT Result / Loser i / Winner i	Truth		Lie	
		Truth	Lie	Truth	Lie
Truth	Truth	$(\alpha_{w_i}, \alpha_{u_i})$	$(\alpha_{w_i}, 0)$	$(\alpha_{w_i} - \alpha^{\otimes}_{w_i} - \gamma, \alpha_{u_i} - \alpha^{\otimes}_{u_i} + \gamma)$	
	Lie	$(0, \alpha_{u_i})$	$(0, 0)$	or $(0, 0)$	
Lie	Truth	$(\alpha^{\oplus}_{w_i} - \alpha_{w_i} - \gamma, \alpha^{\oplus}_{u_i} - \alpha_{u_i} + \gamma)$		$(\alpha^{\oplus}_{w_i} - \alpha^{\otimes}_{w_i} - \gamma, \alpha^{\oplus}_{u_i} - \alpha^{\otimes}_{u_i} + \gamma)$	
	Lie	or $(0, 0)$		or $(0, 0)$	

w_i and v_i represent Winner i and Loser i, respectively. For Player m, α_m, $\alpha^{\oplus}_m$ and $\alpha^{\otimes}_m$ respectively indicate the award for the truth, VT's fake result and DT's lie. As for the value in the bracket, the first value indicates the revenue for Winner i, and the second one indicates the reward for Loser i. For example, $(\alpha_{w_i}, \alpha_{u_i})$ represents the revenue α_{w_i} for Winner i and the reward α_{u_i} for Loser i, respectively. (0, 0) for "DT lies" or "VT lies" means that MVP does not broadcast the match result globally.

Assumption 1 (Identification). *For the game operation, there is a need to identify the key attributes of the match data. In this paper, we assume that under any circumstances (e.g., even if the final results are not consistent), the winners, losers, and MVP can still be identified.*

Next, we must decide whether there is a result supported by a majority of players, so we can upload that match result to the blockchain as a validated result.

Consequently, we can classify the status of a match into *Certainty* and *Confusion*. Definition 3 presents how our model rewards each player under these two statuses.

Definition 3 (Model Design). To begin with, we design the model so that each match is rewarded with a certain amount of virtual coins (i.e., in-game currency). Then, we assign the reward or punishment depending on whether the current status is **Certainty** or **Confusion**.

- **Status = Certainty:** There is only one result supported by the majority of players, which is regarded as the finalized result i.e., consensus. In this case, every player receives the reward with respect to their playing effort during the match. Players providing a result that does not match the consensual one will not be rewarded.
- **Status = Confusion:** None of the results are supported by the majority of players. Then, the MVP decides whether the results should be broadcast. If no, all players will receive nothing (i.e., $\forall i \in TL, \pi_i = 0$). If yes, the MVP will select any match results proposed by the players in their own interest. Let the superscripts $\oplus$ and $\otimes$ distinguish the results provided by the winning team and the losing team(s). α_i indicates the reward of Player i and therefore,
  - $\forall i \in VT, \pi_i = \alpha^{\oplus}_i - \alpha^{\otimes}_i - \gamma$
  - $\forall i \in DT, \pi_i = \alpha^{\oplus}_i - \alpha^{\otimes}_i + \gamma$

By Definition 3, the proposed model is a positive-sum game when all players reach a consensus on the match result, where the system rewards all players for their honesty with freshly minted coins. However, if the system fails to distinguish a valid match result, the mechanism becomes a zero-sum game, where someone's award is from others' loss. In this case, the MVP can certainly attain non-negative revenue, because the MVP will do his/her best in his/her interest, i.e., negative income leads to a deduction from his/her virtual deposit such that he/she does not broadcast the game result. Even if the MVP broadcasts the game results, he/she is not allowed to compete for the next block writer, since the system cannot recognize their actual performance. Note that while most team players are expected to tell the truth, a team player may still tell a lie. In other words, in some cases, a team can have more than one result, and none comes to an agreement. Then, the

MVP will select the best one that maximizes his/her interest. Table 1 shows the payoff matrix of this model.

Note that as assumed at the beginning of Section 3.1, players in a match are not acquainted with one another, making them unlikely to collude. This enforces that a positive-sum game only exists for a consensus that is honest and truthful.

As mentioned before, a positive incentive motivates the MVP to broadcast the game result worldwide. Given a specific bias factor γ, it is a question whether the MVP can obtain a positive bonus when $Status = Confusion$. The following proposition targets releasing the concern by a mathematical formula.

Proposition 1. *Suppose that the schemes of the revenue allocation are uniformly distributed. Given a specific value of γ, which is smaller than $\frac{2n-1}{n(2n+1)}$, the probability μ that the match appears on the blockchain when it does not have an agreement satisfies Equation 2:*

$$\begin{pmatrix} 2(1-\gamma n)nP-(\gamma+1-\gamma n)P+2n-1 \\ 2n-1 \end{pmatrix} = \mu \begin{pmatrix} 2(1-\gamma n)nP+2n-1 \\ 2n-1 \end{pmatrix} + (1-\mu)\begin{pmatrix} 2(1-\gamma n)nP-2(1-\gamma n)P+2n-1 \\ 2n-1 \end{pmatrix} \tag{2}$$

Proof. Under Assumption 1, the total number of revenue allocations is not infinite. With the help of the Generating Function [34], the exact value can be calculated based on (1) the number of cases S_a that the revenues sum up to P for all clients, and (2) the number of invalid cases S_b that the minimum revenue requirement in Definition 1 does not meet. Similarly, we then can acquire the number of cases that MVP will broadcast the unrecognized results (i.e., $Status = Confusion$) according to (3) the number of cases S_c that the MVP receives the reward of more than or equal to a specific value (Note: invalid cases are included in S_c). Therefore, μ is equal to $(S_c - S_b)$ out of $(S_a - S_b)$, i.e., $\mu = \frac{S_c - S_b}{S_a - S_b}$. See Appendix A for full details. □

Proposition 1 illustrates the relation between the value of the biased factor γ and the probability μ that the MVP confirms and broadcasts the result. By calculation and the steps of the proof, the value of μ is always greater than 0. Next, it is necessary to evaluate the existence of $Status = Confusion$.

Nash equilibrium [42] is an important concept in game theory. It is proposed for a non-cooperative game with two or more players to find a balance point, where everyone can obtain an optimal solution regardless of others' decisions. Formally, it is defined as the problem below.

Problem Definition 1 (Nash Equilibrium). *For Player i, there is a special portfolio $\sigma_i^* \in \sum_i$. Let σ_{-i}^* be the situation in which all players select the Nash equilibrium profile other than player i. The full strategy at the equilibrium point can then be written as $\sigma^* = \{\sigma_i^*, \sigma_{-i}^*\}$. The following inequality is always valid.*

$$\forall \sigma_i \in \Sigma_i, \pi_i\left(\sigma_i^*, \sigma_{-i}^*\right) \geq \pi_i\left(\sigma_i, \sigma_{-i}^*\right) \tag{3}$$

The following theorem introduces the Nash equilibrium point of the proposed model.

Theorem 1. *Under Assumption 1, every player tells the truth in Internal Negotiation to obtain the best reward.*

Proof. The core idea of proving Theorem 1 is based on two points: (1) More than half of losers tell the truth, and (2) At least half of winners tell the truth. Therefore, it indicates that $Status = Confusion$ possibly does not exist. Under the circumstance of $Status = Certainty$, liar definitely earns nothing. Hence, the best scenario for every individual is to tell truth. See Appendix B for details. □

ALGORITHM 1: Decentralized Surveillance

Input: Main Chain *mc* and Side Chain *sc*
Predefined function Token(·) ▷ A block's token
Predefined function Reported(·, ·) ▷ Check if an illegal behavior is reported
Predefined function Sum(·, ·) ▷ Summarize an illegal behavior with evidence
illegals = [] ▷ Illegal behaviors
for all (block_*i*, block_*j*) **in** (*mc*, *sc*) **do**
  if Token(block_*i*) == Token(block_*j*) **then**
    if Reported(block_*i*, *mc*) **then**
      illegals.append(Sum(block_*i*, block_*j*))
    end if
  end if
end for
Output: illegals

3.2 Decentralized Surveillance

As previously discussed, it is possible that forked chains are encouraged due to the Nothing-at-stake problem, and this challenges the reputation and thus the usability of the blockchain. Every time a hard fork appears halves the market value of the blockchain system. A practical example would be Peercoin [30], where multiple versions of a chain referring to the same coinbase exist. Similarly, in PoP blockchain, a user may produce blocks on different forks with a token (i.e., an applicable match result). Currently, none of the PoS blockchains suffers a great loss from the Nothing-at-stake problem because the honest nodes in the system tend to refuse to build on forked chains so as to sustain the reputation of the blockchain. Nevertheless, potential vulnerabilities due to forked chains remain an issue that motivates us to address it, e.g., there is a higher risk of double-spending. Here, we present a censorship algorithm inspired by Verus Coin [59] and Casper [8].

Algorithm 1 illustrates the process of finding nodes that attempt to launch malicious attacks due to the Nothing-at-stake problem. It shows that a block is spiteful when its writer uses a match to mine different blocks in different forks. A potential writer can point out any of the attackers in the list in the coming block. If the block is accepted by all nodes, a serious sanction is immediately applied to the cheater (i.e., the award in that block is taken over by the informant and a large fine must be paid). If the adversary fails to pay the penalty, the adversary's account will be blocked from the blockchain system.

Given that users are players that are committed to the game and hence the blockchain system, a community-driven surveillance system would likely be effective even without proper rewards. Yet, if rewards are to be implemented, one must scale the rewards as the amount of minted wealth increases in the game (i.e., inflation). Also, the rewards should be defined in a way that motivates users to report illegal mining (forking chains) within a certain time frame. This way, potential vulnerabilities can be avoided before malicious miners mine too many blocks on a forked chain.

Employing such a censorship algorithm can address both the Nothing-at-stake Problem and the Long-range Attack. For the Nothing-at-Stake problem, as mentioned, an intrinsic reward and a systemic fork identification using Algorithm 1 can encourage "cheat hunters" to hunt down malicious forks, and users will avoid supporting a fork. For the Long-range attack, a malicious fork is difficult to be created with as it requires extraordinary computational capabilities to generate playing efforts (mines), which are frequently hard to forge. Adding to the fact that, similar to the Nothing-at-Stake problem, a community will actively maintain the chain as it is, a malicious fork would find difficulties in being accepted. As such, the Long-range attack is avoided.

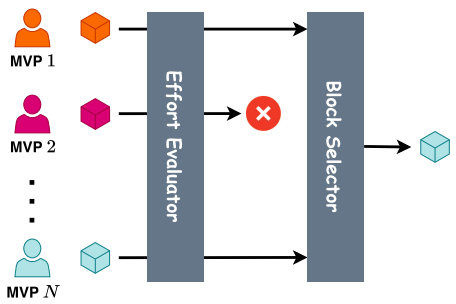


Fig. 3. Block mining.

3.3 Block Mining

As the last step in the whole system, data conservation and synchronization is performed. With the previous two processes, gaming data integrity can be safeguarded and the act of play determined. Finally, we show how an act of play can be evaluated as a step in block mining. Figure 3 introduces how PoP produces a new block. It contains two parts. Effort Evaluator ensures the quality of the players' gaming performance, while Block Selector controls the stability of the confirmation time.

3.3.1 Effort Evaluator. PoP requires almost zero energy (i.e., additional effort) to form an agreement. As a result, there can be a large number of valid producers for the next block, leading to a large number of orphan forks. Hence, limiting the number of players not only reduces internal network congestion, but also prompts players to put in sufficient effort in every competition. One of the ways to determine whether a player is eligible would be to rank this player against all players in terms of their performance (e.g., match scores), however that would be unfair since some players are simply better. So, we decide that self-performance should be the only influential factor, which can be based on whether the final evaluation reaches the threshold and/or whether the player makes progress compared to the past several matches. This way, every player can be a miner as they play the game at their own pace.

Note that in any online game, it is possible that players cannot be paired up with opponents of similar skill level, so a fair effort evaluator could be difficult to define, as players' performance varies. For this, we can assume that the matchmaking process is competent at pairing players of similar skill levels, so that every match is sufficiently fair.

3.3.2 Block Selector. It is universally acknowledged there is only one block at an index. To avoid a large number of collisions, an appropriate formula should be used to limit the number of valid blocks, which controls the confirmation time within a stable and reasonable range. PoP is more energy-friendly as it does not require too many hashing operations. Instead, it uses a lottery-like approach and every time there is a new match, it checks who should be a block writer. Naturally, forks appear when there is more than one valid block. To address this issue, PoP leverages probabilistic finality to determine which one can be infinitely extended.

3.3.3 Finality Discussion. One important issue for a blockchain system is to design a mechanism that confirms whether a mined block should stay in a blockchain. Ideally, a mined block will immediately be broadcasted to all nodes, so that all nodes will start mining the next block. However, practically, a particular block heard by different workers should be at various timestamps due to propagation latency. Therefore, a collision of multiple mined blocks occurs from time to time. Finality describes how many blocks a blockchain should wait for before they are permanently added to the main chain. In relation to this issue, we introduce the following theorem to show

the relationship between when a block is on PoP blockchain and the distribution of gaming lengths.

Theorem 2. *There is a match-based game compromise with normal distribution, whose average gaming length is $\overline{x}$ and the standard deviation is σ. As for a γ confidence level, a permanent block should be confirmed by*

$$\textit{The number of successive blocks} = \left\lceil (\overline{x} + z_{1-\gamma}\sigma)/T \right\rceil \tag{4}$$

where the value of $z_{1-\gamma}$ can be obtained from Z-table [56], T is expected confirmation time and $\lceil a \rceil$ refers to the minimum integer, which is greater than or equal to a.

Proof. Suppose there are two blocks containing the same game results with the same height of different forks. In probabilistic finality, all players in these blocks ought to have bias on which fork should continue excavating such that their reward can be used earlier. We assume the worst case that each fork is supported by 50% of users in the global network. Some players in this block start a new game immediately and become the next block writer. In order to gain the reward from the last block, they need to use the new blocks as votes to decide which one should be on the mainchain to realize a 51% consensus. As a result, other players' influence on which fork is valid becomes trivial. Therefore, it is necessary to determine the maximum average game length representing the majority of players. Since it is difficult to measure using a time difference in a blockchain system, we subsequently leverage the number of confirmations instead. The steps are illustrated as follows:

(1) **Maximum average game length:** According to the given statistics, using the Student's t-distribution [56], a γ confidence interval is constructed as follows for the mean gaming length. With the amount of area in one tail $\alpha = 1 - \gamma$, the level of significance is z_α, which can be attained from Student's t-distribution [56] by assuming that the degree of freedom is infinity. Thus,

$$UCL = \overline{x} + z_\alpha \sigma \tag{5}$$

where UCL is the abbreviation of upper confidence limit, $\overline{x}$ and σ are the average and the standard deviation of the global gaming length, respectively.

(2) **Number of confirmations:** Knowing that the expected time difference between two successive blocks is T, we can find out how to determine a block is on the chain by the confirmation time length and the average time length for block producing. □

3.4 Security Discussion on Cheating Issues

Cheating is a serious issue in the P2P gaming system, which happens via tampering with gaming records or historical transaction records. Without proper disposal, it hinders a game's playability. In this subsection, we list major cheating issues and discuss how our proposed PoP system resolves these challenges.

(1) **Tampering the transaction records:** Adversaries may attempt to alter their game data to provide them as many competitive advantages as possible. In the blockchain system, every interactions that alter the essential game data is recognized by the blockchain system. As a result, there is no way for adversaries to revoke previous records or create invalid records, even if they launch attacks like a long-range attack.

(2) **Changing the game results:** If the gaming records are already on the blockchain, no one can change the records. As for the results pending synchronization, adversaries are not able to reverse, because they cannot modify other players' signatures on the gaming procedures.

(3) **Adding the invalid game results:** Before posting a block on the blockchain, block writers ensure that the game results are valid. Also, if a block contains invalid game results, the users will not accept the block. This indicates that any invalid game results cannot be stored on the blockchain.
(4) **Updating with the forged game results:** Adversaries are able to post forged game results in the phase of Internal Negotiation. However, it is less likely to collude with the players from another team. Therefore, if the players are rational, the game theory model drives the players to disclose the truth and presumably punishes the liars with zero reward.
(5) **Pretending to be multiple players:** This is possible in our proposed blockchain system only when adversaries launch autoplay bots, i.e., an AI system can play the game like a human [65, 68, 69]. As mentioned in Section 3.1, launching the bots to disrupt the blockchain not only is expensive, but also there is very little real-world economic benefit. As a result, majority of the players will tend to be honest, thus keeping the blockchain secured without data tampering, as we mentioned in Section 3.3.

4 NUMERICAL RESULTS AND DISCUSSIONS

4.1 Internal Negotiation

In Section 3.1.1, a game theory model has been proposed to encourage all players to tell the truth. However, behavioral game theory [11] points out that, in reality, people may make a decision contradictory to the rational one. To gain a better understanding of real-life cases, inspired by [20, 43, 60], we conducted an evolutionary game-theoretical simulation to evaluate the trustworthiness of Internal Negotiation.

Our simulation experiment is written in Python. It runs on a PC with a single Intel i7-8550U CPU, a RAM with 16GB and a GPU with NVDIA GeForce MX150. Details of the experimental settings will be described in Section 4.1.1 and the analysis of the simulation will be presented in Section 4.1.2.

4.1.1 Simulation Setup. In order to compromise with Assumption 1, we randomly allocated revenue with a certain percentage to all players. When the final result was ambiguous, the MVP selected and broadcast two different results, one from the victorious team and the other from the defeated team (i.e., based on their benefits). That means, two results, one for the winners and the other for the losers, are deliberately designated such that the information can be successfully recorded in the blockchain. Each player makes a decision to tell the truth or lie.

This simulation is an evolutionary process of players' honest probability[14] and built on the assumption that players are selfish and only make choices to maximize their own revenue. With this assumption, we show that at the steady state, the decisions of all players will eventually converge according to the payoff function, where all players tell the truth.

We simulated the process on 2V2 and 5V5 games, which are common types of match-based games. In each set of simulation experiments, the γ value in the payoff function should first be defined. Next, we define all players as having the same honest probability and who tended to be dishonest at the beginning, where Player m's honest probability was initially denoted as:

$$P_m = init\_pr \tag{6}$$

In each iteration, all players were randomly divided into N matches and performed Internal Negotiation. They needed to make a decision based on a given match result. For any player m in a particular iteration, there are only two decisions (i.e., honesty or dishonesty), and the decision

[14]Honest probability is defined as the chance that a player tells the truth when he/she can earn more in an ambiguous case.

a player makes is solely based on P_m, regardless of the marginal benefit one gains relative to the alternative decision. After all members of a match made a decision, the MVP disclosed the final revenue allocation. Then, for any player m, the proportion of revenue allocation is dis_m and the reward in this match is gain_m, which can be both positive and negative, and the wealth is defined as:

$$Wealth_m = Wealth_m + \text{gain}_m \tag{7}$$

Regardless of the effect of actual income in a 2V2 game, Player m's bonus gain_m is an element of a set $\{1, 0, -1\}$, which respectively refers to positive income (i.e., $\text{dis}_m > 0$), zero income (i.e., $\text{dis}_m = 0$) and negative income (i.e., $\text{dis}_m < 0$). As for a 5V5 game, we assume each match under the condition $Status = Certainty$ mints 100 coins. And we take real income into consideration, where $\text{gain}_m = \text{int}(100 \times \text{dis}_m)$.

If a player gains a negative reward due to being dishonest (i.e., cheating), one will increase the honest probability by a constant value α. If a player receives negative income as a result of being honest, the player will immediately set the honest probability to 0, setting up the player to always be dishonest when confronted with an ambiguous case. Hence, the intermediate honest probability throughout the self-adjustment function can be presented as follows:

$$P_m^* = \begin{cases} \min((1+\alpha)P_m, 1) & \text{gain}_m < 0 \ \& \ \text{Lie} \\ 0 & \text{gain}_m < 0 \ \& \ \text{Truth but can lie} \\ P_m & \text{Otherwise} \end{cases} \tag{8}$$

After self-adjustment, each player further updates their honest probability with reference to the wealth of other players in the same match. This adjustment is necessary for our simulation since players will compare their strategy to that of other players to optimize and thus maximize their reward. The idea is that every player will adjust their honest probability closer to that of those players with more or equal wealth than they have. The way to adjust honest probability in such a way can be represented as below:

$$P_m = \frac{\sum_i P_i^* \cdot Wealth_i \cdot (Wealth_i \geq Wealth_m)}{\sum_i Wealth_i \cdot (Wealth_i \geq Wealth_m)} \tag{9}$$

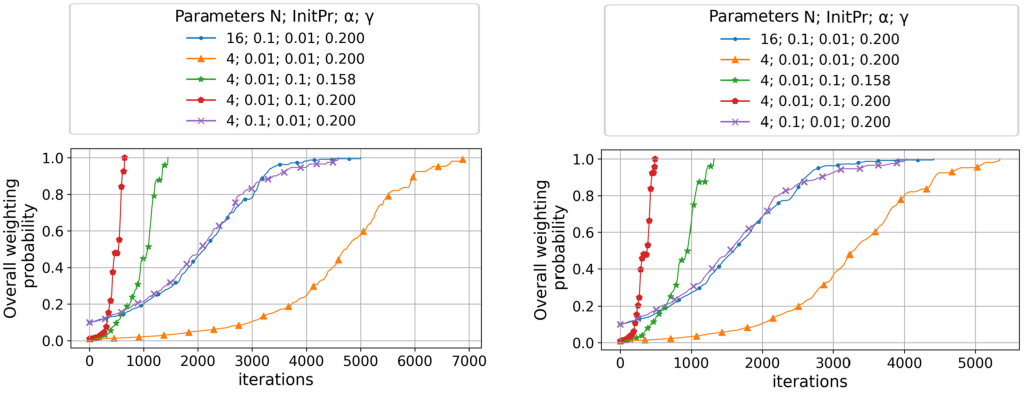
where i indicates the i^{th} player. The iteration of the simulation continues until the system reaches a steady-state, in which all the players stop adjusting their honest probability.

4.1.2 Simulation Result. In this section, we analyze the simulation data from the beginning until the system reaches a steady state. Figure 4 illustrates the results of a 2V2 game, while Figure 5 presents the results of a 5V5 game.

As shown in Figures 4(a) and 5(a), if the number of matches is sufficiently large, both overall weighted probability converge to 1. That means, at steady state, all players will eventually become 100% honest. The result validates the game theory model, showing that the payoff function effectively changes people's minds in real-life cases.

Next, we look at the changing tendency of a player's honest probability from other aspects. Figure 4(b) shows that a player gradually becomes honest with an increasing number of matches earning a positive bonus. In Figure 5(b), it can be seen that a player with more wealth is likely to tell the truth.

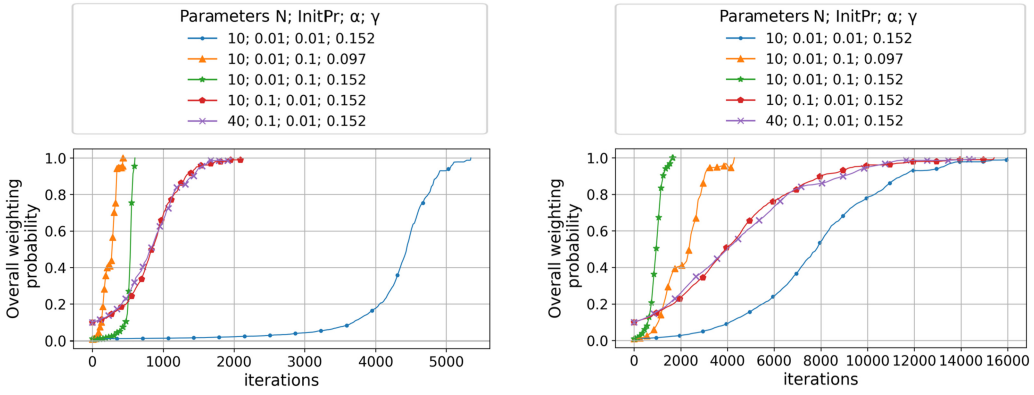
Convergence speed: Although all can reach a steady state, convergence speed is another important factor for consideration. Each iteration means one single match, and a faster convergence rate indicates that players require fewer matches to become honest. As shown in Figures 4(a) and 5(a), the number of players does not influence the final convergence rate. Furthermore, the effect of



(a) Overall Weighted Probability w.r.t. Iterations

(b) Honest probability w.r.t. Number of positive match

Fig. 4. Experiment results from a 2-versus-2 game. For the graph label, the parameters are arranged as follows: N; InitPr; α; γ.



(a) Overall Weighted Probability w.r.t. Iterations

(b) Honest probability w.r.t. Wealth

Fig. 5. Experiment results from a 5-versus-5 game. For the graph label, the parameters are arranged as follows: N; InitPr; α; γ.

alpha is more significant than the effect of the initial probability. The result shows that a flexible player spends less time solidifying the thought that honesty is the best choice in all cases, even if the player lies in the beginning. There is an interesting phenomenon by comparing Figures 4(a) and 5(a). As shown in Figure 4(a), we intuitively hold the view that a larger adjustment value (i.e., gamma) leads to a faster convergence rate because a player is less likely to earn by telling a lie. However, we find a contrary finding in Figure 5(a). In accordance with Figure 5(b), a possible explanation for the unusual phenomenon is that with a large adjustment value, all players receive zero bonus in the early stage, which slows down the convergence speed. In fact, an optimal value may exist for biased factor (i.e., γ) such that the system reaches steady-state with the smallest number of matches.

4.1.3 Empirical Discussion for Cheating on Game Results. This experiment is built on the assumption that players are rational to make a decision and update the lying probability to the best

of their interests. It is noted that the players' rehabilitation process is independent from block mining, although it may affect the average confirmation time, and some uncertain match results can be added to the blockchain. Based on the numerical results, we notice that players will eventually become honest regardless of the initial settings, although the convergence speed differs. In other words, as long as a P2P game is adopting the PoP blockchain, all rational players finally opt to disclose the truth only, notwithstanding that they can obtain a better reward from a fake result.

4.2 Block Mining

We discussed the mining process in Section 3.3. To investigate its performance, we evaluated it experimentally from the perspective of efficiency, fairness and stability. We first introduce the experimental setup in Section 4.2.1. Next, based on the empirical study result, we discuss the advantages of PoP in comparison with PoW in Section 4.2.2.

4.2.1 Experimental Setting. In a highly decentralized blockchain system, a valid block should satisfy the following mathematical formula:

$$\text{SHA256}(\text{SHA256}(\text{Block Kernel})) \leq \text{Target Value} \tag{10}$$

where Block Kernel contains the sensitive data/information of a block (i.e., for protecting data integrity so that data cannot be changed). Note that in PoW, Block Kernel also includes a nonce. For both PoP and PoW, the target value is defined as:

$$\text{Target Value} = \frac{\text{Initial Target Value}}{\text{Difficulty}} \tag{11}$$

where the minimum and initial value of difficulty is 1. In terms of initial target value, PoP is assigned with a fixed constant, while PoW is determined by a range of nonce value, which is written as:

$$\text{Initial Target Value} = \frac{2^{256}}{\text{range of the value of nonce}} \tag{12}$$

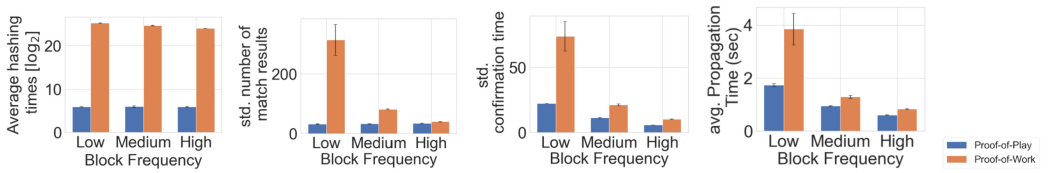
As mentioned in Section 3.3, in PoP, eligible block writers should be selected by the effort evaluator, which is prior to validating a block. In this experiment, the effort evaluator selects eligible MVPs according to their gaming performance (i.e., the grade of a specified match), the threshold of which is dynamic and based on the previous block. A valid block writer should achieve a better performance than the expected one, which is calculated by the average performance of the top 10% of players in the last block. On average, only 5% of MVPs can pass the effort evaluator.

To simulate actual implementation, we collect match data from OpenDota[15] and assume that no players tell lies. We build up the environment with a cluster, which is equipped with OpenMPI, 10 CPU and 400GB RAM. It is assumed that the total number of players is far more than 10. Hence, PoW and PoP use different strategies to tackle inconsistent issues as explained below:

As users/players and miners are separated in PoW, we assume there are only 36 full nodes throughout the global network, which means that only 36 miners sustain the block production. As for PoP, as players and miners are not separated, a distributed system will randomly assign eligible players to one of the background nodes, which neither acts as a transit node nor intervenes in the mining process.

The number of matches is uniformly distributed every second. In each experiment, we only evaluate with 300 successive blocks. The difficulty value will always be 1 in all experiments.

[15] https://www.opendota.com/.



(a) Hashing times (b) avg. propagation time (c) std. matching results (d) std. confirmation time

Fig. 6. Experiment results on Mining Simulation by comparing PoP and PoW. "avg." stands for average while "std." indicates the standard deviation.

4.2.2 Experimental Analysis. For PoW, we vary the block frequency generation by reducing the nonce range, so as to decrease the difficulty value. Although tuning the difficulty value can also affect the poP block producing rate, a more important factor is the number of matches in every second. Based on the block frequency of each experiment, we classify the results into three levels: Low, Medium, and High. For the result of the medium frequency, an experiment on average creates blocks in a range between 0.75 and 1.25 blocks every 12 seconds, while low and high ones respectively refer to two uncovered sides. Figure 6 presents the final results.

Hashing time: Hashing time can indicate how much computational power is consumed in supporting a consensus model. Figure 6(a) shows that with increasing block frequency, PoW gradually declines, while PoP remains relatively stable at a low level. It can be seen that PoP always uses less hashing time to produce a new block. In other words, PoP is more energy efficient than PoW. Furthermore, a valid PoP block can be produced with a constant amount of hashing, showing its stability advantage.

Propagation time: Propagation time refers to the time required from the creation of a block to the entire network receiving it. Figure 6(b) shows that as the block frequency increases, the propagation time of both PoP and PoW goes down. Although their differences become smaller as block frequency rises, the propagation time for PoP is always lower, indicating that the size of a PoP block is smaller than that of a PoW block. In other words, PoP is more efficient in terms of block creation.

Match results: In a PoP system, miners earn bonuses through the synchronization of match results. A fair system should let users receive a similar reward for mining blocks. Considering that the transaction fee of a match is fixed, the standard deviation of the number of matches in a block reflects its fairness. Figure 6(c) indicates that the standard deviation for PoP is quite stable, while the standard deviation for PoW is quite high at low block frequency. Also, the standard deviation of PoP is always smaller than that of PoW, indicating that PoP can perform better than PoW in terms of fairness.

Confirmation time: Confirmation time refers to the time interval between two successive blocks. Hence, the standard deviation of the confirmation time reflects whether blocks are produced on a regular basis. Figure 6(d) shows that the standard deviation of confirmation time for PoW declines dramatically as block frequency increases, while that of PoP decreases slightly. This means that at a higher block frequency, blocks are produced more steadily for both PoP and PoW. As the standard deviation of confirmation time for PoP is always smaller than that for PoW, it indicates that PoP is more stable than PoW in terms of block production.

4.2.3 Numerical Discussion on General Attacks. Next, we investigate the possibility of long-range attack through experiments. To study this important issue, we conducted the following experiments with real-life implementation based on the following assumption. Suppose there are 100 users who play a 2V2 game for an hour. The gaming duration follows a Gaussian Distribution

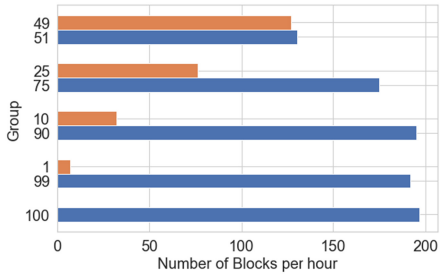


Fig. 7. Various settings of block distribution.

Table 2. Index of when the Chain with More Users is Dominant

Group	1/99	10/90	25/75
Index	1	1	3

with a mean of 120 seconds and a standard deviation of 30 seconds. These players are divided into two groups to mine for different forks. To allow more opportunities for producing a new block, the threshold of Effort Evaluator of the upcoming block is set to the mean of MVPs' performance in the last block. By doing so, about half of the MVPs can become potential block writers.

Figure 7 shows the final length of each fork. Apparently, a group with more users always creates a longer chain than a group with fewer users. The result indicates that an adversary without more than 51% of playing efforts will find it difficult to change the blockchain. Especially in the case of well-known games, long-range attack is almost impossible because an attacker can hardly create a new fork with the required playing efforts and add it to the blockchain.

Apart from security, it is worth discussing the finality issue. Nakamoto [41] and Buterin[16] discuss this issue by making a common assumption that the computational power occupied by an adversary is no more than 25% of total computational power, and the probability of reverting an existing chain is no more than 0.1%. Based on this assumption, Table 2 presents when a chain with more users has the dominant position. Then, by considering Theorem 2 with a confidence level of 99.9%, we can draw a conclusion that a permanent block should be accepted after receiving 12 confirmations. As we can see in Table 2, the empirical result is within the bounds of the theoretical result, indicating that our analysis should be valid.

5 PROTOTYPE IMPLEMENTATION

Infinity Battle [66], an open-source 2V2 turn-based strategy game, is a test bed to show how our model works in a real-life case. We modify this game with communication to our middleware that synchronizes players' data using our PoP-powered blockchain. The game is developed with Unity 2018.2.15f1, while blockchain system is developed with Python 3.

Figure 8 shows the process of integrating PoP with a serverless game. Generally, in terms of a blockchain system, it is divided into two phases – offchain and onchain. At first, a player sets the username and specifies the port number at Login page, while the blockchain system configures this related information in the background. Next, the player enters the welcome page. On this page, one can glance at the heroes, and start playing the game or log out. If a player decides to play the game, a matchmaking node helps the player search for a potential teammate and opponent. If a match

[16] https://blog.ethereum.org/2016/05/09/on-settlement-finality/.

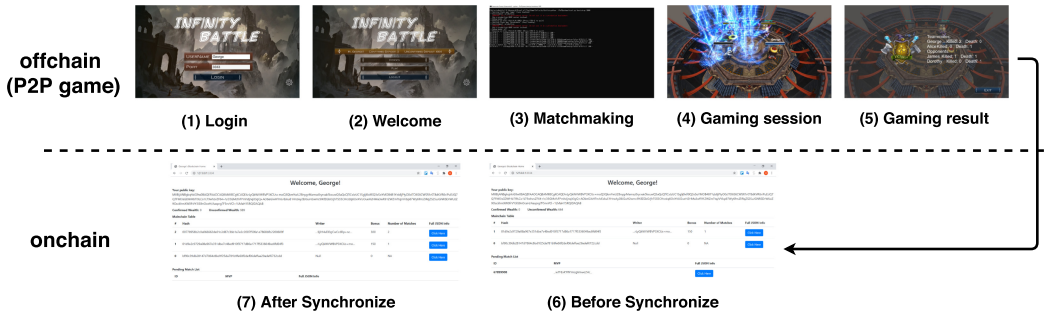


Fig. 8. Procedure of infinity battle.

is successfully created, the player enters the battlefield. After the match ends, the corresponding game result will be processed to store on the blockchain. Based on the PoP consensus model, the corresponding match is included in a new block.

Generally speaking, the game can be divided into three phases, namely **matchmaking**, **gaming session,** and **global synchronization** (i.e., onchain process). The following three subsections elaborate these three components part by part and introduce the technique behind each phase.

5.1 Matchmaking

As the first stage in all multiplayer online games, matchmaking considers numerous factors such as network latency and skill level, and it is the most complex part of a game [1]. To simplify our work on the P2P matchmaking task and preserve the feature of decentralization, we assume that there are a large number of matchmaking nodes all around the world, but there is no connection between any two of them. Consequently, each matchmaking node forms a group only with those players who are connecting to it. In Infinity Battle, there are four players simultaneously connecting to a matchmaking node, and a new match is successfully created. Generally speaking, we assume the matchmaking nodes group up with four comparable players for a new match.

5.2 Gaming Session

Coordinating with the matchmaking node, a group of players form a pure P2P network [53]. They are divided into two teams, and all of them enter the battlefield and start the gaming session. During the gameplay, each player independently broadcasts the actions to all other nodes using the User Datagram Protocol.

Before the battle starts, each player chooses a unique character. Then, in each round, a player can play the game using one of the three predefined attack methods. Although some skills/methods are more powerful, players cannot use all skills and methods in every round. If a character dies, it will be reborn after a certain number of rounds. Furthermore, that character's teammates become stronger before the character is reborn. The game ends when all characters on one team die.

5.3 Global Synchronization

After the gameplay, relevant data will be saved to a distributed storage system based on our proposed model PoP. Since Unity requires to code with C#, PoP using Python provides an API for the game developers to call such that the game results can be broadcast worldwide.

After calling the API from a P2P game, e.g., Infinity Battle, clients launch **Internal Negotiation** and disclose the digital signature on the playing records to other participants. When they receive all the signatures from other participants, they expose the playing records for verification. To

avoid the records modified by some malicious parties, clients post a digital signature based on the collected information. Then, MVP of the match packs up all the information and broadcast it to all active blockchain holders. Alternatively, this process can be done via Gossip protocol [13, 51]. If MVP has remarkable performance in the game, he/she is capable to compete for the next block writer. This is an automatic process, which implements at the backstage and includes **Decentralized Surveillance** and **Block Mining**, and does not hinder the player to start a new match.

6 CONCLUSION

This paper shows a distributed storage system for general P2P games by leveraging blockchain techniques with a consensus model, PoP, which provides data storage functions as well as a guarantee of communication security. It acts as a middleware, containing three core components: Shared Turns, Decentralized Surveillance, and New Block Producing. With these components, PoP is robust enough to secure the data integrity, resist various attacks and uphold the blockchain reputation. We further demonstrated PoP with a game *Infinity Battle* implementation. Compared to three well-known consensus models, PoP has distinct dominant features: it is more efficient, more secure, more decentralized, and fairer.

This paper serves to demonstrate that P2P architecture for online games could be a practical option by leveraging blockchain. With blockchain, the long-envisioned decentralization of online games could finally be achieved without potential cheating and disruptive behaviors. This would be a perfect system, in which developers can finally eliminate the need for server maintenance, without fear of compromising the gaming experience of players; and with the gaming community able to maintain itself using blockchain as a long-lasting and reliable storage overhead, without the fear of termination of developers' support.

APPENDICES

A APPENDIX 1: PROOF OF PROPOSITION 1

According to Definition 3, it is a zero-sum game when $Status = Confusion$. Let $\{\xi_{w_1}, \ldots, \xi_{w_n}, \xi_{u_1}, \ldots, \xi_{u_n}\}$ be the revenue distribution among all players, where $\xi_i = \alpha_i^{\oplus} - \alpha_i^{\otimes}$. By Definition 1, every player can obtain a minimum proportion reward of $\gamma/2$. There are $2n$ players throughout the whole system and consequently, the maximum proportion is $(1 - \frac{\gamma}{2}(2n-1))$ for a player's revenue. As a result, the range of the distribution is between $\frac{\gamma}{2}$ and $(1 - \frac{\gamma}{2}(2n-1))$, which means $\xi_i \in (\gamma n - 1, 1 - \gamma n)$. Specifically, MVP's income should be greater than $1/2n$ because MVP's revenue is higher than the averaging level. According to Definition 3, MVP has a chance to broadcast the result when $\xi_{MVP} > \gamma$. We consider the maximum interval for MVP and therefore, the applicable range for the biased factor γ should be:

$$1 - \frac{\gamma}{2}(2n-1) - \frac{1}{2n} > \gamma \implies \gamma < \frac{2n-1}{n(2n+1)} \tag{13}$$

A common practice to determine the number of eligible cases is by means of Generating Function [34]. This method requires the use of positive integers. According to the boundary of ξ_i and the precision P (Note: for the satisfaction of the minimum unit of the system), we can convert $\{\xi_{w_1}, \ldots, \xi_{w_n}, \xi_{u_1}, \ldots, \xi_{u_n}\}$ into positive integers: Let $\zeta_i = (\xi_i + 1 - \gamma n)P$, such that $\zeta_i \in [0, 2(1-\gamma n)P)$. Therefore, we need to determine the number of solutions to the following equation:

$$\zeta_{w_1} + \cdots + \zeta_{w_n} + \zeta_{u_1} + \cdots + \zeta_{u_n} = 2(1 - 2\gamma n)nP \tag{14}$$

We first compute the number of schemes when $\zeta_i \geq 0$. The number is equal to the coefficient of $x^{(2(1-\gamma n)nP)}$ in the generating function:

$$f(x) = (1 + x^1 + x^2 + \ldots)^{2n} \tag{15}$$

Thus, based on the generic methodology in [34], the solution to it should be:

$$S_a := \binom{2(1-\gamma n)nP + 2n - 1}{2n - 1} \tag{16}$$

As mentioned before, there is a boundary for ζ_i. Consequently, the case is invalid when a ζ_i that is larger than or equal to $2(1-\gamma n)P$ exists. Likewise, the number of invalid cases should be:

$$S_b := \binom{2(1-\gamma n)nP - 2(1-\gamma n)P + 2n - 1}{2n - 1} \tag{17}$$

In order to find someone's revenue that is greater than or equal to γ, which means $\zeta_i \geq (\gamma + 1 - \gamma n)P$. Likewise, the number of corresponding cases can be found as follows:

$$S_c := \binom{2(1-\gamma n)nP - (\gamma + 1 - \gamma n)P + 2n - 1}{2n - 1} \tag{18}$$

Considering the constraints of the biased factor γ, we know that the magnitude among S_a, S_b and S_c should be: $S_a > S_c > S_b$. With S_a, S_b and S_c, there are a total of $(S_a - S_b)$ valid cases, among which only $(S_c - S_b)$ cases can synchronize with the blockchain system. We denote μ as a probability that MVP obtains a positive revenue. Based on the statement above, we have:

$$\frac{S_c - S_b}{S_a - S_b} = \mu \tag{19}$$

B APPENDIX 2: PROOF OF THEOREM 1

According to Equation (13), MVP possibly broadcasts the result if and only if the bias factor is smaller than $\frac{2n-1}{n(2n+1)}$. In other words, if the bias factor is greater than or equal to $\frac{2n-1}{n(2n+1)}$, the best scenario is to tell the truth. Next, we analyze our model under the condition that

$$\gamma < \frac{2n-1}{n(2n+1)}$$

First, let us consider $Status = Certainty$ (i.e., VT Result = Truth and DT Result = Truth). According to Table 1, none of the rational players will tell a lie if this scenario occurs. Next, the following discusses the case $Status = Confusion$, where none of the results receive a majority agreement.

Lemma 1. *More than half of losers tell the truth.*

Proof. According to Table 1, the reason a loser (i.e., u_i) may cheat the system is that a player could possibly earn more benefit in telling a lie. In other words, one of the following two inequalities holds:

$$\alpha_{u_i} < \alpha_{u_i} - \alpha_{u_i}^{\otimes} + \gamma \quad \text{or} \quad \alpha_{u_i}^{\oplus} - \alpha_{u_i} + \gamma < \alpha_{u_i}^{\oplus} - \alpha_{u_i}^{\otimes} + \gamma \tag{20}$$

Since the sum of all players' proportion is 1, the total proportion of the VT reduction is consistent with the total rate of the DT increment. With Definition 2 and Assumption 1, a rational player is confident about lying because one sees a satisfying result of $\alpha_{MVP}^{\otimes} \leq \alpha_{MVP} - \gamma$. On account for

$$\frac{\gamma + \frac{n-1}{2}\gamma}{\gamma} > \frac{n}{2}; \quad \frac{\gamma + \frac{n-1}{2}\gamma}{n} > \frac{1}{2}\gamma$$

we can draw a conclusion that more than half of losers obtain an increment of $\gamma/2$. In other words, more than half of losers' revenue in a loser's lie possess

$$\alpha^{\otimes}_{u_i} > \max(\gamma, \alpha_{u_i}),$$

which cannot satisfy any inequalities in (20) since it requires $\alpha^{\otimes}_{u_i} < \max(\gamma, \alpha_{u_i})$. □

From Lemma 1, $Status = Confusion$ possibly holds only when a majority of the players tell lies. The following lemma explores whether the requirement is possible.

Lemma 2. *At least half of winners state the truth.*

Proof. We analyze the statement on a case-by-case basis.

(1) VT Result = Truth, DT Result = Lie: If a winner (i.e., w_i) tells a lie, according to Table 1, one should satisfy:

$$\alpha_{w_i} < \alpha_{w_i} - \alpha^{\otimes}_{w_i} - \gamma$$

The inequality impossibly holds because the left hand side is always greater than the right hand side. This means that all winners will tell the truth.

(2) VT Result = Lie, DT Result = Truth: Let us assume there are $\lceil (n+1)/2 \rceil$ winners lying. A winner liar should be motivated by $\alpha_{w_i} < \alpha^{\oplus}_{w_i} - \alpha_{w_i} - \gamma$ and $\alpha_{MVP} \leq \alpha^{\oplus}_{MVP} - \gamma$. Under Assumption 1, we have

$$\begin{aligned}
&\alpha^{\oplus}_{w_1} + \cdots + \alpha^{\oplus}_{w_n} + \alpha^{\oplus}_{u_1} + \cdots + \alpha^{\oplus}_{u_n} \\
&> (2\alpha_{w_1} + \gamma) + \cdots + \left(2\alpha_{w_{\lceil \frac{n+1}{2} \rceil}} + \gamma\right) + \left(\alpha_{w_{\lceil \frac{n+1}{2} \rceil + 1}} + \frac{\gamma}{2}\right) + \cdots + \left(\alpha_{w_n} + \frac{\gamma}{2}\right) \\
&\quad + (\alpha_{u_1} - \gamma) + \cdots + (\alpha_{u_n} - \gamma) \\
&= (\alpha_{w_1} + \cdots + \alpha_{w_n} + \alpha_{u_1} + \cdots + \alpha_{u_n}) + \left(\alpha_{w_1} + \cdots + \alpha_{w_{\lceil \frac{n+1}{2} \rceil}}\right) \\
&\quad + \left\lceil \frac{n+1}{2} \right\rceil \gamma + \left(n - \left\lceil \frac{n+1}{2} \right\rceil\right) \frac{\gamma}{2} - n\gamma \\
&\overset{(a)}{>} 1 + \left\lceil \frac{n+1}{2} \right\rceil \gamma - \frac{n\gamma}{2} \geq 1 + \frac{(n+1)\gamma}{2} - \frac{n\gamma}{2} = 1 + \frac{\gamma}{2} > 1
\end{aligned} \tag{21}$$

where (a) is based on two facts: (I) $\alpha_{w_1} + \cdots + \alpha_{w_n} + \alpha_{u_1} + \cdots + \alpha_{u_n} = 1$ and (II) the minimum proportion of reward is $\gamma/2$. Knowing that $\alpha^{\oplus}_{w_1} + \cdots + \alpha^{\oplus}_{w_n} + \alpha^{\oplus}_{u_1} + \cdots + \alpha^{\oplus}_{u_n} = 1$, here raises a conflict and therefore, the statement that at most half of all winners tell a lie is valid.

(3) VT Result = Lie, DT Result = Lie: Similar to the previous one, we first assume there are $\lceil (n+1)/2 \rceil$ winners lying. A winner opting to lie in this situation is because of $\alpha_{w_i} < \alpha^{\oplus}_{w_i} - \alpha^{\otimes}_{w_i} - \gamma$ and $\alpha^{\otimes}_{MVP} \leq \alpha^{\oplus}_{MVP} - \gamma$. Therefore,

$$\begin{aligned}
\left(\alpha^{\oplus}_{w_1} - \alpha^{\otimes}_{w_1}\right) + \cdots + \left(\alpha^{\oplus}_{w_n} - \alpha^{\otimes}_{w_n}\right) &> \left(\frac{\gamma}{2} + \gamma\right) \cdot \left\lceil \frac{n+1}{2} \right\rceil + \left(n - \left\lceil \frac{n+1}{2} \right\rceil\right) \frac{\gamma}{2} \\
&\geq \frac{3}{2} \left\lceil \frac{n+1}{2} \right\rceil \gamma + \left(n - \left\lceil \frac{n+1}{2} \right\rceil\right) \frac{\gamma}{2} > \gamma n
\end{aligned} \tag{22}$$

As mentioned before, the percentage increase in VT is equal to the percentage decrease in DT. With Assumption 1, $(\alpha^{\oplus}_{w_1} - \alpha^{\otimes}_{w_1}) + \cdots + (\alpha^{\oplus}_{w_n} - \alpha^{\otimes}_{w_n}) \leq \gamma n$, which means that our assumption is contradictory to the real case. Therefore, at least half of the winners should tell the truth.

□

From Lemmas 1 and 2, a conclusion can be drawn that more than half of players tell the truth. As a result, the match status should be certain, and those who cheat the system will eventually

receive nothing. To maximize personal income, players should tell the truth even if their expected income in telling a lie is higher. Therefore, there is only one Nash equilibrium point, which is

$$\sigma^* = \left(\sigma_{w_1}^{(t)}, \sigma_{w_2}^{(t)}, \ldots, \sigma_{w_n}^{(t)}, \sigma_{u_1}^{(t)}, \sigma_{u_2}^{(t)}, \ldots, \sigma_{u_n}^{(t)}\right) \tag{23}$$

REFERENCES

[1] Sharad Agarwal and Jacob R. Lorch. 2009. Matchmaking for online games and other latency-sensitive P2P systems. In *ACM SIGCOMM Computer Communication Review*, Vol. 39. ACM, 315–326.

[2] Adam Back et al. 2002. Hashcash-a denial of service counter-measure. (2002).

[3] Ignasi Barri, Concepció Roig, and Francesc Giné. 2016. Distributing game instances in a hybrid client-server/P2P system to support MMORPG playability. *Multimedia Tools and Applications* 75, 4 (2016), 2005–2029.

[4] Ashwin Bharambe, John R. Douceur, Jacob R. Lorch, Thomas Moscibroda, Jeffrey Pang, Srinivasan Seshan, and Xinyu Zhuang. 2008. Donnybrook: Enabling large-scale, high-speed, peer-to-peer games. *SIGCOMM Comput. Commun. Rev.* 38, 4 (Aug. 2008), 389–400. https://doi.org/10.1145/1402946.1403002

[5] Michał Boroń, Jerzy Brzeziński, and Anna Kobusińska. 2020. P2P matchmaking solution for online games. *Peer-to-Peer Networking and Applications* 13, 1 (2020), 137–150.

[6] Jonah Brown-Cohen, Arvind Narayanan, Alexandros Psomas, and S. Matthew Weinberg. 2019. Formal barriers to longest-chain proof-of-stake protocols. In *Proceedings of the 2019 ACM Conference on Economics and Computation*. 459–473.

[7] Vitalik Buterin et al. 2014. *Ethereum White Paper: A Next Generation Smart Contract & Decentralized Application Platform. First version* (2014).

[8] Vitalik Buterin and Virgil Griffith. 2017. Casper the friendly finality gadget. *arXiv preprint arXiv:1710.09437* (2017).

[9] Wei Cai, Zehua Wang, Jason B. Ernst, Zhen Hong, Chen Feng, and Victor C. M. Leung. 2018. Decentralized applications: The blockchain-empowered software system. *IEEE Access* 6 (2018), 53019–53033.

[10] Wei Cai and Xiao Wu. 2019. Demo abstract: An interoperable avatar framework across multiple games and blockchains. In *IEEE INFOCOM 2019-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 967–968.

[11] Colin F. Camerer. 2011. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press.

[12] E. Deirmentzoglou, G. Papakyriakopoulos, and C. Patsakis. 2019. A survey on long-range attacks for proof of stake protocols. *IEEE Access* 7 (2019), 28712–28725. https://doi.org/10.1109/ACCESS.2019.2901858

[13] Alan Demers, Dan Greene, Carl Hauser, Wes Irish, John Larson, Scott Shenker, Howard Sturgis, Dan Swinehart, and Doug Terry. 1987. Epidemic algorithms for replicated database maintenance. In *Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing*. 1–12.

[14] Jesse Donkervliet, Animesh Trivedi, and Alexandru Iosup. 2020. Towards supporting millions of users in modifiable virtual environments by redesigning minecraft-like games as serverless systems. In *12th {USENIX} Workshop on Hot Topics in Cloud Computing (HotCloud 20)*.

[15] Ali Dorri, Salil S. Kanhere, Raja Jurdak, and Praveen Gauravaram. 2019. LSB: A lightweight scalable blockchain for IoT security and anonymity. *J. Parallel and Distrib. Comput.* 134 (2019), 180–197.

[16] John R. Douceur. 2002. The Sybil attack. In *Peer-to-Peer Systems*, Peter Druschel, Frans Kaashoek, and Antony Rowstron (Eds.). Springer Berlin, Berlin, 251–260.

[17] Scott Douglas, Egemen Tanin, Aaron Harwood, and Shanika Karunasekera. 2005. Enabling massively multi-player online gaming applications on a P2P architecture. In *Proceedings of the IEEE International Conference on Information and Automation*. 7–12.

[18] Haihan Duan, Jiaye Li, Sizheng Fan, Zhonghao Lin, Xiao Wu, and Wei Cai. 2021. Metaverse for social good: A university campus prototype. In *Proceedings of the 29th ACM International Conference on Multimedia*. 153–161.

[19] Stefan Dziembowski, Sebastian Faust, Vladimir Kolmogorov, and Krzysztof Pietrzak. 2015. Proofs of space. In *Annual Cryptology Conference*. Springer, 585–605.

[20] David Easley, Jon Kleinberg, et al. 2012. Networks, crowds, and markets: Reasoning about a highly connected world. *Significance* 9 (2012), 43–44.

[21] Lu Fan, Hamish Taylor, and Phil Trinder. 2007. Mediator: A design framework for P2P MMOGs. In *Proceedings of the 6th ACM SIGCOMM Workshop on Network and System Support for Games*. 43–48.

[22] Lu Fan, Phil Trinder, and Hamish Taylor. 2010. Design issues for peer-to-peer massively multiplayer online games. *International Journal of Advanced Media and Communication* 4, 2 (2010), 108–125.

[23] Geoffrey Fox. 2001. Peer-to-peer networks. *Computing in Science & Engineering* 3, 3 (2001), 75–77.

[24] John S. Gilmore and Herman A. Engelbrecht. 2011. A survey of state persistency in peer-to-peer massively multiplayer online games. *IEEE Transactions on Parallel and Distributed Systems* 23, 5 (2011), 818–834.

[25] Seyoung Huh, Sangrae Cho, and Soohyung Kim. 2017. Managing IoT devices using blockchain platform. In *2017 19th International Conference on Advanced Communication Technology (ICACT)*. IEEE, 464–467.
[26] EOS IO. 2018. EOS. IO Technical White Paper v2. *EOS, Tech. Rep., March* (2018).
[27] Patric Kabus and Alejandro P. Buchmann. 2007. Design of a cheat-resistant P2P online gaming system. In *Proceedings of the 2nd International Conference on Digital Interactive Media in Entertainment and Arts*. ACM, 113–120.
[28] Sukrit Kalra, Rishabh Sanghi, and Mohan Dhawan. 2018. Blockchain-based real-time cheat prevention and robustness for multi-player online games. In *Proceedings of the 14th International Conference on Emerging Networking Experiments and Technologies*. 178–190.
[29] Kostis Karantias, Aggelos Kiayias, and Dionysis Zindros. 2020. Proof-of-burn. In *International Conference on Financial Cryptography and Data Security*. Springer, 523–540.
[30] Sunny King and Scott Nadal. 2012. PPCoin: Peer-to-peer crypto-currency with proof-of-stake. *self-published paper, August* 19 (2012).
[31] Markus Klems, Jacob Eberhardt, Stefan Tai, Steffen Härtlein, Simon Buchholz, and Ahmed Tidjani. 2017. Trustless intermediation in blockchain-based decentralized service marketplaces. In *International Conference on Service-Oriented Computing*. Springer, 731–739.
[32] B. Knutsson, Honghui Lu, Wei Xu, and B. Hopkins. 2004. Peer-to-peer support for massively multiplayer games. In *IEEE INFOCOM 2004*, Vol. 1. 107. https://doi.org/10.1109/INFCOM.2004.1354485
[33] Daniel Kraft. 2016. Game channels for trustless off-chain interactions in decentralized virtual worlds. *Ledger* 1, 0 (2016), 84–98. https://doi.org/10.5195/ledger.2016.15
[34] Gregory Levitin et al. 2005. *The Universal Generating Function in Reliability Analysis and Optimization*. Vol. 6. Springer.
[35] Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. 2020. A survey on the security of blockchain systems. *Future Generation Computer Systems* 107 (2020), 841–853.
[36] Iuon-Chang Lin and Tzu-Chun Liao. 2017. A survey of blockchain security issues and challenges. *IJ Network Security* 19, 5 (2017), 653–659.
[37] Huey-Ing Liu and Yun-Ting Lo. 2008. DaCAP-a distributed Anti-Cheating peer to peer architecture for massive multiplayer on-line role playing game. In *2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*. IEEE, 584–589.
[38] Mitar Milutinovic, Warren He, Howard Wu, and Maxinder Kanwal. 2016. Proof of luck: An efficient blockchain consensus protocol. In *Proceedings of the 1st Workshop on System Software for Trusted Execution*. 1–6.
[39] Tian Min and Wei Cai. 2019. A security case study for blockchain games. In *2019 IEEE Games, Entertainment, Media Conference (GEM)*. IEEE, 1–8.
[40] Tian Min, Hanyi Wang, Yaoze Guo, and Wei Cai. 2019. Blockchain games: A survey. *arXiv preprint arXiv:1906.05558* (2019).
[41] Satoshi Nakamoto et al. 2008. Bitcoin: A peer-to-peer electronic cash system. (2008).
[42] John F. Nash et al. 1950. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences* 36, 1 (1950), 48–49.
[43] Jonathan Newton. 2018. Evolutionary game theory: A renaissance. *Games* 9, 2 (2018), 31.
[44] Giang-Truong Nguyen and Kyungbaek Kim. 2018. A survey about consensus algorithms used in blockchain. *Journal of Information Processing Systems* 14, 1 (2018).
[45] Oscar Novo. 2018. Blockchain meets IoT: An architecture for scalable access management in IoT. *IEEE Internet of Things Journal* 5, 2 (2018), 1184–1195.
[46] Md. Mehedi Hassan Onik and Mahdi H. Miraz. 2019. Performance analytical comparison of blockchain-as-a-service (BaaS) platforms. In *International Conference for Emerging Technologies in Computing*. Springer, 3–18.
[47] Martin J. Osborne et al. 2004. *An Introduction to Game Theory*. Vol. 3. Oxford University Press New York.
[48] Andrew Paradise and Dennis Zografos. 2016. Integrations portal for a peer-to-peer game platform. US Patent 9,349,246.
[49] Sunoo Park, Krzysztof Pietrzak, Joël Alwen, Georg Fuchsbauer, and Peter Gazi. 2015. *SpaceCoin: A Cryptocurrency Based on Proofs of Space (Vol. 528)*. Technical Report. IACR Cryptology ePrint Archive. https://eprint.iacr.org/2015/528.pdf.
[50] Jared N. Plumb and Ryan Stutsman. 2018. Exploiting Google's edge network for massively multiplayer online games. In *2018 IEEE 2nd International Conference on Fog and Edge Computing (ICFEC)*. IEEE, 1–8.
[51] Gokay Saldamli, Charit Upadhyay, Devika Jadhav, Rohit Shrishrimal, Bapugouda Patil, and Lo'ai Tawalbeh. 2022. Improved gossip protocol for blockchain applications. *Cluster Computing* (2022), 1–12.
[52] Mayra Samaniego, Uurtsaikh Jamsrandorj, and Ralph Deters. 2016. Blockchain as a service for IoT. In *2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*. IEEE, 433–436.
[53] R. Schollmeier. 2001. A definition of peer-to-peer networking for the classification of peer-to-peer architectures and applications. In *Proceedings First International Conference on Peer-to-Peer Computing*. 101–102. https://doi.org/10.1109/P2P.2001.990434

[54] Eric Setton, Jeonghun Noh, and Bernd Girod. 2006. Low latency video streaming over peer-to-peer networks. In *2006 IEEE International Conference on Multimedia and Expo*. IEEE, 569–572.
[55] Sushil Kumar Singh, Shailendra Rathore, and Jong Hyuk Park. 2020. BlockIoTIntelligence: A blockchain-enabled intelligent IoT architecture with artificial intelligence. *Future Generation Computer Systems* 110 (2020), 721–743.
[56] Student. 1908. The probable error of a mean. *Biometrika* (1908), 1–25.
[57] Daniel Stutzbach and Reza Rejaie. 2006. Understanding churn in peer-to-peer networks. In *Proceedings of the 6th ACM SIGCOMM Conference on Internet Measurement*. 189–202.
[58] BUFF Team. 2018. BUFF: Game for fun, earn for real. (2018).
[59] Michael J. Toutonghi, Michael F. Toutonghi, and Alex R. English. 2018. Verus Coin. (2018).
[60] Karl Tuyls and Ann Nowé. 2005. Evolutionary game theory and multi-agent reinforcement learning. *The Knowledge Engineering Review* 20, 1 (2005), 63–90.
[61] Tengfei Wang, Shuyi Zhang, Xiao Wu, and Wei Cai. 2019. Rhythm Dungeon: A blockchain-based music roguelike game. In *Proceedings of the 14th International Conference on the Foundations of Digital Games*. 1–3.
[62] Yao Wang and Julita Vassileva. 2003. Trust and reputation model in peer-to-peer networks. In *Proceedings Third International Conference on Peer-to-Peer Computing (P2P2003)*. IEEE, 150–157.
[63] Steven Daniel Webb, Sieteng Soh, and Jerry L. Trahan. 2009. Secure referee selection for fair and responsive peer-to-peer gaming. *Simulation* 85, 9 (2009), 608–618.
[64] Nxt Wiki. 2019. Whitepaper:Nxt — Nxt Wiki,. https://nxtwiki.org/index.php?title=Whitepaper:Nxt&oldid=53653. [Online; accessed 21-September-2019].
[65] Bin Wu. 2019. Hierarchical macro strategy model for MOBA game AI. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 1206–1213.
[66] Feijie Wu, Ho Yin Yuen, Henry C. B. Chan, Victor C. M. Leung, and Wei Cai. 2020. Infinity Battle: A glance at how blockchain techniques serve in a serverless gaming system. In *Proceedings of the 28th ACM International Conference on Multimedia* (Seattle, WA, USA) *(MM'20)*. Association for Computing Machinery, New York, NY, USA, 4559–4561. https://doi.org/10.1145/3394171.3414458
[67] Amir Yahyavi and Bettina Kemme. 2013. Peer-to-peer architectures for massively multiplayer online games: A survey. *ACM Computing Surveys (CSUR)* 46, 1 (2013), 9.
[68] Deheng Ye, Guibin Chen, Wen Zhang, Sheng Chen, Bo Yuan, Bo Liu, Jia Chen, Zhao Liu, Fuhao Qiu, Hongsheng Yu, et al. 2020. Towards playing full MOBA games with deep reinforcement learning. *Advances in Neural Information Processing Systems* 33 (2020), 621–632.
[69] Deheng Ye, Guibin Chen, Peilin Zhao, Fuhao Qiu, Bo Yuan, Wen Zhang, Sheng Chen, Mingfei Sun, Xiaoqian Li, Siqin Li, et al. 2020. Supervised learning achieves human-level performance in MOBA games: A case study of Honor of Kings. *IEEE Transactions on Neural Networks and Learning Systems* (2020).
[70] Ho Yin Yuen, Feijie Wu, Wei Cai, Henry C. B. Chan, Qiao Yan, and Victor C. M. Leung. 2019. Proof-of-play: A novel consensus model for blockchain-based peer-to-peer gaming system. In *Proceedings of the 2019 ACM International Symposium on Blockchain and Secure Critical Infrastructure*. 19–28.
[71] Nairan Zhang, Youngki Lee, Meera Radhakrishnan, and Rajesh Krishna Balan. 2015. GameOn: P2P gaming on public transport. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. 105–119.
[72] Wenbing Zhao, Shunkun Yang, and Xiong Luo. 2019. On consensus in public blockchains. In *Proceedings of the 2019 International Conference on Blockchain Technology*. 1–5.
[73] Zibin Zheng, Shaoan Xie, Hong-Ning Dai, Xiangping Chen, and Huaimin Wang. 2018. Blockchain challenges and opportunities: A survey. *International Journal of Web and Grid Services* 14, 4 (2018), 352–375.

Received 24 June 2021; revised 17 May 2022; accepted 29 August 2022