

# 1 AC-BlockDFL: Audit-driven Committee BlockDFL for Secure Federated Learning

## 2

### 3 ANONYMOUS AUTHOR(S)

#### 4

5 Blockchain-based Federated Learning (BCFL) faces a critical scalability-security trade-off. While committee-based architectures  
6 significantly reduce communication overhead, they introduce a fundamental vulnerability: the *Progressive Committee Capture Attack*  
7 (PCCA). In PCCA, rational adversaries exploit the stake-election-reward feedback loop to gradually capture committee control through  
8 strategic starvation and stake accumulation. We propose *AC-BlockDFL*, a defense framework that decouples system security from  
9 committee honesty through optimistic execution and asynchronous auditing. By internalizing the externalities of malicious behavior  
10 via a game-theoretic slashing protocol, AC-BlockDFL ensures that attacks yield negative expected utility. Our evaluation over 2,000  
11 rounds demonstrates that AC-BlockDFL suppresses malicious stake ratios from  $1.3\times$  to  $0.37\times$ , reducing unavailability rates from 22.3%  
12 to below 1% while maintaining  $O(C^2)$  communication complexity.

#### 13

14 CCS Concepts: • Security and privacy → Distributed systems security; • Computing methodologies → Artificial intelligence.

#### 15

16 Additional Key Words and Phrases: Federated Learning, Blockchain, Committee Consensus, Game Theory, Incentive Compatibility

#### 17

#### 18 ACM Reference Format:

19 Anonymous Author(s). 2024. AC-BlockDFL: Audit-driven Committee BlockDFL for Secure Federated Learning. *J. ACM* 37, 4, Article 111  
20 (August 2024), 20 pages. <https://doi.org/XXXXXXX.XXXXXXX>

#### 21

## 22 1 Introduction

#### 23

24 Blockchain-based Federated Learning (BCFL) has emerged as a promising paradigm for collaborative machine learning  
25 in environments where mutual trust among participants cannot be assumed. Real-world deployments in Low Earth  
26 Orbit (LEO) satellite networks [8, 24, 32], vehicular networks (V2X) [16, 25], and Industrial IoT [17, 27] demonstrate  
27 compelling use cases where decentralized coordination is essential. In LEO constellations, for instance, ground station  
28 contact windows last merely five minutes with downlink bandwidth limited to approximately 8 Mbps [32], rendering  
29 centralized aggregation architectures impractical. BCFL addresses these constraints by establishing decentralized trust  
30 infrastructure across heterogeneous satellite operators, reducing model convergence time by up to thirty hours [8].

#### 31

32 However, BCFL systems face a fundamental scalability bottleneck when approaching large-scale deployment. The  
33 predominant use of Practical Byzantine Fault Tolerance (PBFT) [4] and its variants introduces  $O(N^2)$  message complexity,  
34 causing consensus latency to dominate training time as participant counts grow. Empirical measurements from  
35 FLCoin [28] reveal that at 100 nodes, single-round consensus generates over 20,000 message exchanges with latency  
36 exceeding 25 seconds—comparable to or exceeding the model training duration itself. Storage requirements compound  
37 this challenge: Bitcoin full nodes require approximately 200 GB while Ethereum exceeds 465 GB, fundamentally  
38 incompatible with edge devices possessing only KB-to-MB scale memory [1].

#### 39

40 Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not  
41 made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components  
42 of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on  
43 servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [permissions@acm.org](mailto:permissions@acm.org).

44 © 2024 Copyright held by the owner/author(s). Publication rights licensed to ACM.

45 Manuscript submitted to ACM

46 Manuscript submitted to ACM

53 *The Committee Mechanism.* To address these scalability constraints, recent work has converged on *committee-based*  
54 architectures that delegate verification responsibility to a smaller subset of validators. Selection mechanisms include hash-  
55 ring sampling [26], stake-weighted election [15, 28], and Verifiable Random Function (VRF) based sortition [12, 29, 31].  
56 These approaches yield substantial efficiency gains: FLCoin [28] reports 90% communication overhead reduction and  
57 5.7× training speedup through sliding-window election, while BFLC [15] achieves sub-three-second consensus latency.  
58 Effectively, committee mechanisms reduce communication complexity from  $O(N^2)$  to  $O(C^2)$  or even  $O(C)$ , where  
59  $C \ll N$  is the committee size.  
60

62 *The Blind Spot.* Despite these advances, existing BCFL literature harbors a critical yet overlooked vulnerability:  
63 the implicit assumption that committee members remain honest or that the proportion of malicious nodes stays  
64 static throughout system operation. Current defenses focus predominantly on data-plane attacks—Byzantine-robust  
65 aggregation rules such as Krum, Trimmed Mean, and Median [3, 34] assume an honest majority among aggregating  
66 nodes. However, these mechanisms provide no protection when the committee itself becomes compromised. As  
67 FedBlock [20] observes, when any participant may become a validator, systems cannot rely solely on honest majority  
68 assumptions but must actively detect and isolate malicious verifiers—a capability conspicuously absent from current  
69 BCFL architectures.  
70

73 *The PCCA Threat.* We identify and formalize a novel attack vector: the *Progressive Committee Capture Attack* (PCCA).  
74 Unlike direct Byzantine attacks, PCCA adversaries employ *strategic starvation*—upon gaining committee control,  
75 attackers prioritize processing their own model updates while systematically denying service to honest participants.  
76 This manipulation of the reward distribution mechanism enables attackers to “legitimately” accumulate stake over  
77 successive rounds, progressively cementing their dominance until decentralized governance collapses entirely. Crucially,  
78 once the honest majority assumption fails in any given round, existing systems lack mechanisms to identify or penalize  
79 malicious actors, allowing attackers to maintain their advantage indefinitely.  
80

83 *Contributions.* This paper presents *Audit-driven Committee BlockDFL* (AC-BlockDFL), a defense framework that  
84 decouples system security from collective committee honesty through optimistic execution with asynchronous auditing.  
85 Our contributions are threefold:  
86

- 88 (1) **Attack Formalization.** We provide the first formal definition of the Progressive Committee Capture Attack  
89 (PCCA) and a rational adversary model that captures strategic, incentive-driven behavior. Through systematic  
90 simulation, we quantify PCCA’s destructive impact on long-term incentive compatibility.
- 92 (2) **AC-BlockDFL Architecture.** We propose a novel defense architecture combining optimistic execution with  
93 asynchronous auditing. A distributed challenger network performs post-hoc verification of committee decisions,  
94 enabling detection and penalization of fraudulent aggregation results even when the committee is fully  
95 compromised.
- 97 (3) **Game-Theoretic Guarantees.** We design an internal slashing protocol grounded in game-theoretic analysis,  
98 ensuring that auditing costs remain strictly below potential gains from malicious behavior. We prove that  
99 honest participation constitutes the unique Nash equilibrium under repeated play [6]. Extensive simulations  
100 over 2,000 rounds demonstrate that AC-BlockDFL maintains model accuracy above 98.6% under 30% adversarial  
101 collusion, reduces communication overhead by 44.4% at equivalent security levels, and suppresses minimum  
102 unavailability rate from 20% to below 5%.

## 105    2 Background

106 This section establishes the theoretical foundations and technical background necessary for understanding the security  
 107 challenges in committee-based blockchain federated learning. We first discuss the fundamental trust dilemma in  
 108 federated learning, then introduce the principles of Byzantine fault tolerance, and finally establish the baseline system  
 109 model for committee-based architectures.

### 112    2.1 Federated Learning and the Trust Dilemma

114 Federated learning (FL) represents a paradigm shift in distributed machine learning, encapsulating the principle of  
 115 “bringing the model to the data” rather than aggregating data centrally [18]. While FL significantly enhances data  
 116 privacy by locally constraining raw data, its standard architecture relies on a fundamental assumption: participants  
 117 must trust a central aggregation server to honestly execute aggregation and uniformly distribute results.

119 In the absence of verifiable consistency, the central server constitutes a single point of failure and a primary  
 120 vulnerability. A malicious or compromised server could perform selective aggregation, intentionally excluding specific  
 121 updates, or directly tamper with the global model to inject backdoors [2, 10]. Furthermore, while FL avoids direct data  
 122 transmission, a malicious aggregator can still infer sensitive information from client updates [11, 36]. This trust dilemma  
 123 severely restricts the deployment of FL in high-value, cross-organizational scenarios where participating entities may  
 124 be independent or competitive, necessitating a decentralized trust infrastructure [13, 17].

### 127    2.2 Byzantine Fault Tolerance Fundamentals

129 Blockchain technology, characterized by immutability, transparency, and decentralization, serves as an ideal infrastruc-  
 130 ture to resolve the FL trust dilemma. However, the security of blockchain fundamentally relies on consensus protocols  
 131 designed to tolerate malicious behavior, rooted in the Byzantine Generals Problem [14].

133 The mathematical constraint of Byzantine Fault Tolerance (BFT) dictates that a system of  $N$  nodes can tolerate  
 134 at most  $f$  malicious nodes, requiring  $N \geq 3f + 1$ . This one-third threshold originates from the *quorum intersection*  
 135 *principle*: to ensure sufficient honest endorsements, any decision needs  $2f + 1$  confirmations. The intersection of any  
 136 two  $2f + 1$  sets guarantees the inclusion of at least  $f + 1$  nodes, meaning at least one honest node witnesses both  
 137 decisions, preventing contradictory states.

139 Practical Byzantine Fault Tolerance (PBFT) [4] reduces the communication complexity of BFT consensus to  $O(N^2)$   
 140 through a three-phase commit protocol (Pre-prepare, Prepare, Commit). While efficient for small networks, the quadratic  
 141 communication cost becomes a severe bottleneck for large-scale FL systems requiring frequent iterations involving  
 142 hundreds of participants.

### 145    2.3 Committee-based BCFL Architecture

147 To reconcile the efficiency demands of federated learning with the security requirements of blockchain, the *committee-*  
 148 *based architecture* has emerged as the prevailing design paradigm. By delegating consensus responsibilities to a smaller,  
 149 representative subset of nodes (the committee), these systems reduce communication complexity from  $O(N^2)$  to  
 150  $O(C^2 + N)$  where the committee size  $C \ll N$  [15, 20, 28].

152    *Baseline System Model: BlockDFL..* We adopt BlockDFL [26] as our baseline system model, representing the state-of-  
 153 the-art in peer-to-peer BCFL. BlockDFL implements role separation, partitioning participants into three distinct roles  
 154 per training round:

- 157 (1) **Update Providers:** Execute local model training on private data and submit bounded updates.  
 158 (2) **Aggregators:** Collect updates, perform filtering, and compute the aggregated global proposal.  
 159 (3) **Validators:** Form the committee to evaluate competing proposals via Krum scoring [3] and execute PBFT  
 160 consensus to select the final model.  
 161

162  
 163 *The Stake-Election-Reward Cycle.* Role assignment in BlockDFL relies on *stake-weighted deterministic random selection*,  
 164 using the previous block’s hash as an unpredictable entropy source mapping onto a stake-weighted hash ring. This  
 165 creates a critical economic incentive structure designed to solve the free-rider problem:  
 166

- 167 • **Stake:** Determines election probability; participants with higher stake bounds are proportionally more likely to  
 168 be selected as Aggregators or Validators.  
 169
- 170 • **Reward:** Distributed exclusively to contributors of the accepted proposal (the winning Aggregator, included  
 171 Update Providers, and Validators who voted for it).  
 172

173 This mechanism instantiates a *positive feedback loop*: receiving rewards increases absolute stake, which enhances future  
 174 election probability and aggregation weight, thereby amplifying the likelihood of subsequent rewards. While intended  
 175 to cultivate long-term honest contributions, this very dynamic introduces vulnerabilities to strategic persistence.  
 176

### 177 3 Related Work

178 The convergence of federated learning and blockchain technology has precipitated diverse architectural innovations  
 179 to address decentralized coordination, privacy, and security [21]. Early systems like DeepChain [31] and Biscotti [29]  
 180 focused on preserving privacy during aggregation using cryptographic commitments and differential privacy. However,  
 181 scaling BFT consensus to accommodating hundreds of FL clients remained a persistent challenge due to its  $O(N^2)$   
 182 communication overhead [16, 27].  
 183

#### 184 3.1 Evolution of Committee Architectures

185 To mitigate consensus bottlenecks, recent literature has pivoted toward committee-based designs inspired by Algorand’s  
 186 sortition [12]. By randomly selecting a constant-size committee to perform consensus, systems effectively decouple  
 187 performance from total network size. FLCoin [28] utilized a sliding-window mechanism based on contribution history  
 188 to form dynamic committees, achieving up to 90% reduction in communication overhead. Similarly, BFLC [15] adopted  
 189 a reputation-based election scheme, prioritizing nodes with high historical quality scores. Recent works such as  
 190 FedBlock [20] and RapidChain [35] further explore sharding and adaptive committee selection to optimize efficiency.  
 191

192 While optimization successes are evident, these election mechanisms inherently couple system security to committee  
 193 composition. If an adversary captures a supermajority (e.g.,  $> 2/3$ ) of the committee seats, traditional data-layer defenses  
 194 like Krum [3] or Trimmed Mean [34] are entirely bypassed because the compromised committee itself executes these  
 195 algorithms.  
 196

#### 197 3.2 Limitations of Existing Verification Methods

198 Addressing Byzantine behavior in decentralized aggregation currently relies on three primary verification paradigms:  
 199

200     *Cryptographic Verification (zkML).* Zero-Knowledge Machine Learning (zkML) provides the strongest security  
 201 guarantees by compiling learning computations into arithmetic circuits, allowing verification without re-execution [5],  
 202 Manuscript submitted to ACM

[37], [30]. However, zkML faces prohibitive computational bottlenecks [33]. Compiling simple architectures like ResNet-18 generates millions of polynomial constraints, and generating proofs takes minutes. More critically, zkML currently cannot support complex, non-linear Byzantine-robust aggregation algorithms like Krum, which require  $O(N^2 \cdot d)$  pairwise distance calculations that cause circuit explosions [9].

*Optimistic Execution (opML).*.. Optimistic Machine Learning (opML) defaults to accepting computations but allows a challenge window during which "AnyTrust" challengers can submit fraud proofs via interactive bisection protocols [7], [23]. While efficient, mainstream opML architectures natively resolving disputes on-chain require challenge periods extending up to a week (e.g., Optimism [22]). These latencies fundamentally conflict with the high-frequency iterative nature of federated learning, rendering opML broadly inapplicable for per-round FL aggregation.

*Committee Consensus and Static Blindspots.* Committee-based verification remains the most pragmatic solution but is underpinned by static probabilistic assumptions: current security analyses calculate committee capture probability by assuming a fixed adversarial population distribution [15, 28]. This static view completely overlooks the behavior of *rational, strategic adversaries* [6, 19]. As indicated by our analysis of the stake-election-reward cycle, systems like BlockDFL [26] and FedBlock [20] contain positive feedback loops. Strategic attackers can exploit this by behaving honestly ("lurking") to accumulate stake and reputation until they acquire sufficient influence to capture the committee ("starving" honest participants) [6].

### 3.3 Our Contribution

Our work directly addresses the systemic blindspot in static committee security. We define and analyze the Progressive Committee Capture Attack (PCCA), demonstrating how rational adversaries bypass conventional defenses. In contrast to existing static committee systems, AC-BlockDFL breaks the stake feedback loop through optimistic execution coupled with an asynchronous, stake-slashing auditing layer, securing the system economically while preserving the efficiency benefits of committee architectures.

## 4 Threat Model and Security Analysis

The security of AC-BlockDFL rests on mitigating sophisticated, economically-driven vulnerabilities while maintaining committee-scale efficiency. In this section, we formalize the adversary model, define the Progressive Committee Capture Attack (PCCA), and establish the dual-layer security guarantees and game-theoretic incentive compatibility of our framework.

### 4.1 Threat Model and Adversary Capabilities

Unlike traditional Byzantine fault tolerance research that assumes purely destructive behavior, our threat model considers a *Rational Adversary* whose primary objective is long-term economic utility maximization and governance control. This aligns with realistic blockchain incentive structures.

*Capabilities.* The adversary controls a fraction of the network nodes, denoted by  $f$ , where typically  $f \leq 0.3$ . These malicious nodes are not isolated; they can collude, coordinate voting strategies, and share information. Crucially, the adversary is highly strategic: malicious nodes can perfectly emulate honest behavior to build reputation and accumulate resources during early stages, instantly switching to malicious actions when an advantageous opportunity arises.

261 Furthermore, the adversary has full visibility into the public blockchain state, including stake distributions and historical  
 262 committee components.  
 263

264 *Limitations.* The adversary is constrained by standard cryptographic assumptions (e.g., unforgeable digital signatures,  
 265 collision-resistant hash functions) and cannot tamper with immutable historical records on the blockchain. Furthermore,  
 266 the adversary cannot command an absolute network majority (e.g., > 50%) due to the prohibitively high capital costs.  
 267 Finally, their actions are governed by economic rationality; they will not execute attacks where the expected financial  
 268 penalty strictly outweighs the potential gains.  
 269

#### 270 4.2 Progressive Committee Capture Attack (PCCA)

271 Current BlockDFL defenses overwhelmingly focus on data-plane attacks (e.g., data poisoning or model replacement),  
 272 relying heavily on robust aggregation algorithms like Krum [3] or Trimmed Mean [34]. However, these algorithms  
 273 implicitly operate under a critical “Validator Blind Spot”: they assume that the aggregator or committee executing the  
 274 algorithm is inherently honest. We formalize the *Progressive Committee Capture Attack (PCCA)*, a consensus-plane  
 275 vulnerability that exploits the positive feedback loop of stake-based committee selections: “Stake → Election → Reward  
 276 → Stake”. By subverting the committee at the consensus layer, PCCA enables attackers to entirely bypass data-plane  
 277 defenses, rendering robust aggregation mathematically irrelevant if the nodes evaluating the models are themselves  
 278 malicious. The attack unfolds in two distinct phases:  
 279

280 *Phase 1: Lurking (Shadow Mode).* Because committee selection relies on stake-weighted random sampling, gaining a  
 281 majority requires significant capital or chance. In this phase, the adversary strictly adheres to protocol rules—submitting  
 282 high-quality model updates and verifying proposals honestly. This patience allows the adversary to accumulate baseline  
 283 stake and evade anomaly detection. The adversary waits for a serendipitous election window where malicious nodes  
 284 coincidentally obtain a supermajority (e.g., > 2/3) of the seats in a single committee.  
 285

286 *Phase 2: Occupying (Capture Mode).* Upon securing a committee supermajority, the adversary drops the honest facade.  
 287 The specific execution depends on the alignment of the current round’s aggregator:  
 288

- 289 • **Strategic Starvation:** If the aggregator is honest, the malicious committee executes a denial-of-service by  
 290 systematically voting against valid, high-quality proposals. Consequently, honest aggregators and data providers  
 291 are denied their block rewards. This starvation stunts the stake growth of honest participants, mathematically  
 292 guaranteeing that the adversary captures a disproportionately larger share of the systemic inflation. Over  
 293 successive rounds, this inflates the adversary’s probability of being selected for future committees.  
 294
- 295 • **Full Stack Poisoning:** If the aggregator is also controlled by the adversary, they achieve “full-stack control.”  
 296 The malicious aggregator deliberately accepts poisoned model updates (e.g., backdoor triggers or flipped labels),  
 297 and the malicious committee force-approves the proposal. This directly degrades the global model’s accuracy  
 298 while monopolizing the round’s rewards.  
 299

300 Through PCCA, the adversary’s relative stake advantage over honest nodes dynamically shifts. However, this equity  
 301 growth does not diverge infinitely. Even under severe Strategic Starvation, honest Update Providers whose local data  
 302 aligns with the malicious consensus still receive foundational data provision rewards, whereas honest Validators and  
 303 Aggregators are starved of their larger operational rewards. Consequently, the adversary’s stake fraction converges to a  
 304 steady-state advantage limit  $\lim_{t \rightarrow \infty} \frac{S_{mal}(t)}{S_{hon}(t)} = \alpha \cdot \frac{S_{mal}(0)}{S_{hon}(0)}$ , where the advantage coefficient  $\alpha$  typically bounds between  
 305 1.1 ~ 1.25 depending on the reward distribution ratio. While mathematically bounded, this ≈ 20% artificial equity  
 306  
 307  
 308  
 309  
 310  
 311  
 312 Manuscript submitted to ACM

313 premium is devastating in practice: it permanently elevates the adversary’s probability of winning future committee  
 314 seats, locking the system into a stable but persistent governance imbalance that transforms a decentralized network  
 315 into a malicious oligarchy.

316     *Differentiating PCCA from Traditional BFT Attacks.* It is crucial to distinguish PCCA from classic Byzantine fault  
 317 tolerance (BFT) attack vectors. Traditional attacks (e.g., 51% attacks or eclipse attacks) are typically *irrational* or  
 318 *destructive*, aiming to cause immediate network forks or halt consensus liveness entirely, which makes them highly  
 319 visible. In contrast, PCCA is a *rational, stealthy* governance attack. The adversary does not seek to crash the network;  
 320 rather, they aim to silently monopolize the economic flow (mining block rewards) while maintaining normal blockchain  
 321 liveness. By strategically starving honest nodes, the attack is indistinguishable from standard protocol execution from a  
 322 structural standpoint, requiring economic security countermeasures rather than pure cryptographic or network-layer  
 323 defenses.  
 324

### 325     4.3 Security Guarantees

326 Our security model comprises two layers with distinct trust assumptions designed specifically to break the PCCA  
 327 feedback loop: a *detection layer* requiring only a single honest challenger, and an *arbitration layer* leveraging standard  
 328 Byzantine fault tolerance.

329     *Detection Layer: 1-of-N Honest Assumption.* The detection layer operates under an exceptionally weak assumption:  
 330 among all  $N$  network participants, at least one honest node is willing to act as a challenger. This is substantially weaker  
 331 than the  $2/3$  honest majority required by traditional BFT systems, as it requires only the *existence* of a single honest  
 332 participant rather than coordinated action by a majority. The feasibility of this assumption stems from blockchain’s  
 333 transparency—all proposal CIDs are recorded on-chain with corresponding data publicly accessible via IPFS, enabling  
 334 any node to independently verify committee decisions.

335     **THEOREM 4.1 (DETECTION COMPLETENESS).** *Let  $\mathcal{V}_r$  denote the verification committee for round  $r$ ,  $\text{Krum}(\{p_a\})$  be  
 336 the deterministic correct result of executing Krum over all aggregation proposals, and  $w_{r+1}$  be the global update actually  
 337 committed by the committee. If  $w_{r+1} \neq \text{Krum}(\{p_a\})$  and there exists at least one honest node  $c^*$  among all  $N$  participants  
 338 willing to act as challenger, then this deviation is necessarily detected.*

339     **PROOF.** The proof relies on Krum’s determinism and the public verifiability of on-chain data. Given identical inputs  
 340  $\{p_a\}$ , any executor obtains the unique output  $\text{Krum}(\{p_a\})$  regardless of identity or location. Since all proposal CIDs are  
 341 recorded on-chain during committee consensus and corresponding data is accessible via IPFS, the honest challenger  $c^*$   
 342 can: (1) retrieve the identical input set  $\{p_a\}$  from IPFS, (2) independently execute Krum locally to obtain  $\text{Krum}(\{p_a\})$ ,  
 343 and (3) compare against the committed  $w_{r+1}$ . Any discrepancy constitutes verifiable proof of deviation, enabling  $c^*$  to  
 344 submit a valid challenge transaction. Since verification depends solely on publicly accessible on-chain CIDs, IPFS data,  
 345 and deterministic computation, the committee cannot evade detection through information hiding or ambiguity.  $\square$

346     *Arbitration Layer: Global 2/3 Honest Assumption.* When a challenge is initiated, adjudication authority transfers from  
 347 the committee to the entire network under standard BFT assumptions: honest nodes must exceed  $2/3$  of total nodes, i.e.,  
 348  $N_{\text{total}} > 3f$  where  $f$  is the number of Byzantine nodes. During arbitration, all validators download relevant proposals  
 349 via IPFS, re-execute Krum, and vote on challenge validity through PBFT consensus.

365 THEOREM 4.2 (PUNISHMENT CERTAINTY). Let  $N_{\text{total}}$  be the total network nodes with Byzantine count  $f$  satisfying  
 366  $N_{\text{total}} > 3f$ . If a challenger successfully detects committee misbehavior per Theorem 4.1 and submits a valid challenge  
 367 transaction, then the misbehavior is necessarily confirmed during arbitration, and all colluding committee members suffer  
 368 complete stake slashing.

370 PROOF. Upon challenge submission, the smart contract retrieves all proposal CIDs for the disputed round and  
 371 triggers network-wide re-verification. By Krum’s determinism, all honest validators compute identical correct results  
 372  $\text{Krum}(\{p_a\})$  and can determine whether  $w_{r+1}$  deviates. Under  $N_{\text{total}} > 3f$ , at least  $N_{\text{total}} - f > 2N_{\text{total}}/3$  honest nodes  
 373 participate in arbitration voting. These honest nodes, based on identical deterministic computation, unanimously vote  
 374 to confirm the deviation. Since PBFT requires  $> 2/3$  agreement and honest nodes exceed this threshold, arbitration  
 375 consensus necessarily succeeds. The smart contract then automatically executes predefined slashing logic, confiscating  
 376 the full stake of all committee members who endorsed the deviant result. This execution is guaranteed by smart contract  
 377 determinism and immune to external interference.  $\square$

381 Theorems 4.1 and 4.2 jointly establish the complete security logic: the former ensures misbehavior is *necessarily*  
 382 *discovered*, the latter ensures discovered misbehavior is *necessarily punished*. This dual certainty forms the logical  
 383 foundation for economic security.

#### 385 4.4 Cost of Attack

387 We now formalize the capital threshold an attacker must surpass to execute a profitable attack while evading punishment.  
 388 Two distinct barriers must be overcome: (1) probabilistically winning  $> 2/3$  committee seats via random election, and  
 389 (2) deterministically controlling  $\geq 1/3$  of network voting power to block arbitration.

391 THEOREM 4.3 (ATTACK COST LOWER BOUND). In AC-BLOCKDFL, an attacker seeking to execute a malicious committee  
 392 decision while completely evading economic punishment must control stake capital satisfying:

$$\text{Cost}_{\text{total}} \geq \frac{1}{3} N \cdot \bar{s} \quad (1)$$

396 where  $N$  is the total network size and  $\bar{s}$  is the average stake per node. Even with this capital, the attacker must still  
 397 probabilistically obtain  $> 2/3$  committee seats through random election.

399 PROOF. Achieving “successful attack without punishment” requires overcoming two security layers. At the committee  
 400 level, the attacker must obtain  $> 2/3$  validator seats in the target round’s random election—a probabilistic event  
 401 determined by stake proportion that cannot be made certain. At the network level, per Theorem 4.2, once a challenge is  
 402 initiated and verified, all malicious stakes are fully slashed. To evade punishment, the attacker must control  $\geq [N/3]$  of  
 403 network voting power to break arbitration liveness by preventing PBFT consensus. This deterministic capital threshold  
 404 scales linearly with network size as  $O(N)$ . Since punishment evasion is a logical prerequisite for profitable attack, the  
 405 total attack cost lower bound is  $\frac{1}{3} N \cdot \bar{s}$ .  $\square$

409 This theorem reveals a fundamental security amplification: AC-BLOCKDFL’s asynchronous audit mechanism elevates  
 410 the economic barrier from committee-scale  $O(C)$  to network-scale  $O(N)$ . In traditional BlockDFL without post-hoc  
 411 accountability, attack cost depends solely on controlling a small committee. AC-BLOCKDFL forces attackers to first  
 412 solve the problem of countering a  $2/3$  honest network majority before mounting any concrete attack. Given typical  
 413 deployments where  $N \gg C$  (e.g.,  $N = 100$ ,  $C = 7$  in our experiments), this layered defense provides substantial  
 414 robustness.

417  
418  
419  
420  
421  
422  
423  
424  
Table 1. Attacker payoff matrix under AC-BLOCKDFL

Strategy	Gain	Loss (if detected)
Honest behavior	$R_{\text{round}}$ (proportional)	0
Attack (success)	$\leq 7.0$ units	500 units

424  
425  

#### 4.5 Game-Theoretic Analysis

426  
427  
We employ game-theoretic analysis to demonstrate that honest behavior constitutes the unique Nash equilibrium for  
428  
all rational participants under AC-BLOCKDFL’s economic mechanism.429  
430  
431  
432  
*Attacker Payoff Model.* For a rational attacker, the decision problem can be modeled as expected payoff computation  
in a single-shot game. Let  $G_{\text{attack}}$  denote the maximum single-round gain from controlling the committee, and  $L_{\text{slash}}$  the  
stake loss from full slashing. The expected payoff is:

433  
434  
$$E[\text{Payoff}] = P_{\text{success}} \cdot G_{\text{attack}} - P_{\text{caught}} \cdot L_{\text{slash}} \quad (2)$$

435  
436  
437  
438  
439  
where  $P_{\text{success}}$  is the probability of controlling the committee in a given round, and  $P_{\text{caught}}$  is the probability of detection  
and punishment. Note that  $P_{\text{success}}$  measures election outcomes while  $P_{\text{caught}}$  measures detection probability—distinct  
events at different layers. The attacker can only mount an attack when winning the election; once attacked, Theorems 4.1–  
4.2 ensure  $P_{\text{caught}} \rightarrow 1$  under our dual-layer assumptions. Thus:

440  
441  
$$E[\text{Payoff}] = P_{\text{success}} \cdot (G_{\text{attack}} - L_{\text{slash}}) \quad (3)$$

442  
443  
444  
445  
446  
447  
*Incentive Compatibility Condition.* The sufficient condition for incentive compatibility emerges clearly: whenever  
 $L_{\text{slash}} > G_{\text{attack}}$ , expected payoff is strictly negative regardless of  $P_{\text{success}}$ . Under our endogenous staking model,  $L_{\text{slash}} =$   
 $\lambda \times R_{\text{round}}$  while  $G_{\text{attack}}$  is upper-bounded by  $C \times R_{\text{round}}$  (monopolizing all validation rewards). Since  $\lambda \gg C$  by design,  
this condition holds stably independent of token market fluctuations.448  
449  
450  
451  
452  
*Numerical Analysis.* Using our experimental parameters: committee size  $C = 7$ , per-validator reward 1.0 units, initial  
stake 100 units. Maximum single-round gain  $G_{\text{attack}} \leq 7.0$  units. Upon detection, at least 5 colluding members each lose  
their full 100-unit stake, yielding  $L_{\text{slash}} = 500$  units—approximately 71× the potential gain.453  
454  
455  
456  
This extreme risk-reward asymmetry ensures negative expected payoff even under optimistic attacker assumptions.  
For  $E[\text{Payoff}] < 0$ , we require  $P_{\text{caught}} > G_{\text{attack}}/L_{\text{slash}} \approx 1.4\%$ . Our security theorems guarantee  $P_{\text{caught}} \rightarrow 1$ , far  
exceeding this minimal threshold.457  
458  
459  
460  
461  
462  
*Nash Equilibrium.* Honest behavior constitutes the unique Nash equilibrium: no rational player can improve their  
payoff by unilaterally deviating to attack. The slashing mechanism breaks the positive feedback loop enabling gradual  
committee capture attacks [6]—instead of accumulating stake through manipulation, attackers suffer substantial stake  
reduction, permanently eliminating their governance influence. This equilibrium remains stable across all token market  
conditions due to the endogenous stake pricing design.463  
464  

## 5 AC-BlockDFL System Design

465  
466  
467  
Traditional Byzantine fault-tolerant (BFT) consensus mechanisms provide robust security guarantees but their  $O(N^2)$   
communication complexity inherently conflicts with the highly iterative nature of federated learning. As analyzed in

469 Section 4.1, existing committee-based approaches mitigate this overhead but introduce severe structural vulnerabilities:  
470 small committees are susceptible to progressive stake accumulation, and their reliance on honest-majority assumptions  
471 fails to deter rational, strategic attackers.  
472

473 To break this deadlock, we introduce Audit-driven Committee BlockDFL (AC-BlockDFL). This framework builds  
474 upon the baseline BlockDFL model but fundamentally shifts the paradigm from “threshold security” to “economic  
475 security” through asynchronous auditing and internal slashing protocols.  
476

### 477 5.1 Design Philosophy

478 The core design philosophy of AC-BlockDFL stems from re-evaluating the relationship between a blockchain’s state  
479 finality and the iterative training characteristics of federated learning. Financial transaction systems demand immediate,  
480 irrevocable correctness for every transaction because any error could lead to permanent asset loss; this forces traditional  
481 blockchains to achieve network-wide consensus *before* any state change. However, federated learning operates through  
482 multiple rounds of iterative refinement. A single round’s deviation can be naturally corrected by subsequent training  
483 rounds.  
484

485 This observation creates the design space to decouple synchronous execution from correctness verification. AC-  
486 BlockDFL adopts an *optimistic execution* philosophy: it allows the system to immediately commit model updates once the  
487 small committee reaches consensus, while rigorous correctness verification is deferred to a non-blocking, asynchronous  
488 background audit. Security is maintained not by preventing malicious behavior upfront, but by ensuring that any  
489 attempt to manipulate committee consensus faces an economic penalty that exponentially exceeds any potential gain,  
490 thereby eliminating the economic incentive for attacks in a game-theoretic setting.  
491

### 492 5.2 System Architecture and Workflow

493 AC-BlockDFL preserves the foundational training workflow of BlockDFL—including the localized training by Update  
494 Providers, proposal generation by Aggregators, and Krum-based [3] scoring via PBFT by Validators. However, it  
495 introduces three critical architectural extensions:  
496

- 497 (1) **The Challenger Role:** A fourth participant role subject to open-access principles. Any network node willing  
498 to stake the required deposit can act as a Challenger for a given round. This ensures supervisory power  
499 remains highly decentralized. Challengers continuously monitor on-chain records, independently re-execute  
500 the deterministic Krum algorithm, and compare their results against the committee’s committed global update.  
501
- 502 (2) **Off-chain Storage Integration:** To prevent ledger bloat, AC-BlockDFL offloads the heavy model gradients to  
503 the InterPlanetary File System (IPFS), recording only Content Identifiers (CIDs) and metadata on-chain. This  
504 reduces on-chain storage complexity from  $O(\text{ModelSize})$  to  $O(\text{HashSize})$ . To ensure data availability during  
505 audits, nodes pin the relevant IPFS data for the duration of a defined Challenge Window.  
506
- 507 (3) **Execute-then-Audit Paradigm:** Unlike BlockDFL where committee decisions are final without recourse,  
508 AC-BlockDFL commits the model immediately but simultaneously opens an asynchronous audit window. This  
509 retains high liveness while preserving the ability to trigger a network-wide arbitration if anomalies are detected,  
510 effectively democratizing oversight to the entire network.  
511

512 *Instant Update Protocol.* Algorithm 1 formalizes the protocol’s “happy path.” During Phase 1, role assignment is  
513 deterministically computed by all nodes based on stake-weighted randomness derived from the previous block hash.  
514 In Phase 2, aggregators upload aggregated proposals to IPFS and submit the resulting CIDs on-chain. In Phase 3, the  
515 Manuscript submitted to ACM  
516

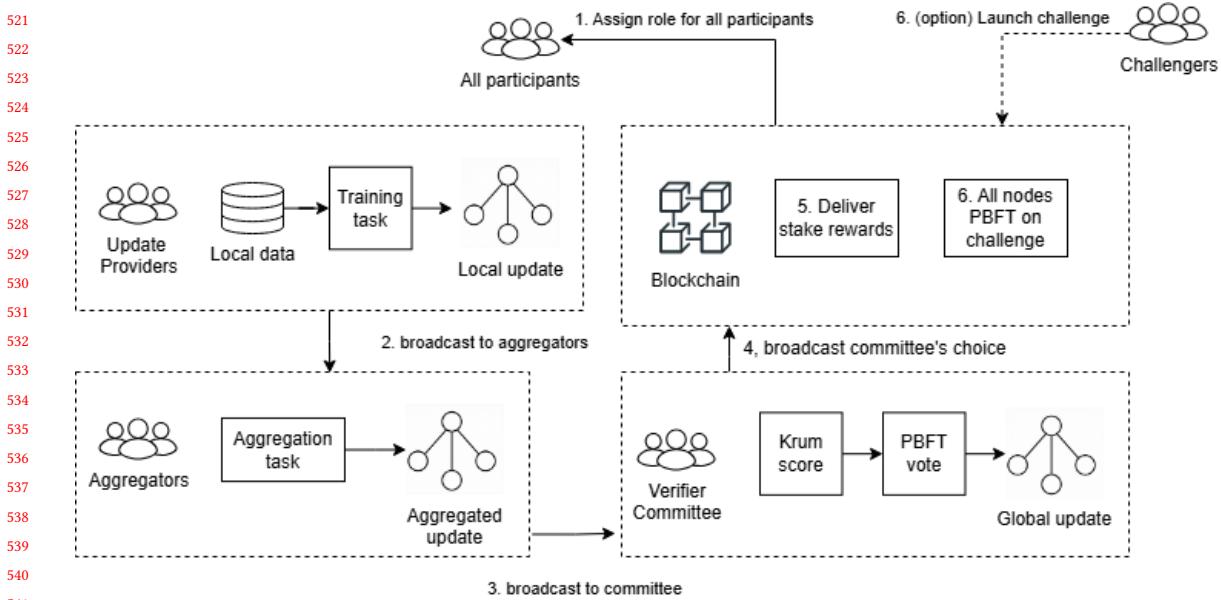


Fig. 1. AC-BlockDFL System Architecture and Workflow. The committee performs optimistic execution while the Challenger network performs asynchronous auditing.

---

**Algorithm 1** AC-BlockDFL Execution Protocol (Instant Update)
 

---

**Require:** Round  $r$ , Total Stake Weighted Nodes  $\mathcal{N}$

**Ensure:** Updated Global Model  $w_{r+1}$

- 1: **Phase 1 – Role Assignment:** Select  $\mathcal{V}, \mathcal{A}, \mathcal{U} \subset \mathcal{N}$  via stake-weighted randomness.
  - 2: **Phase 2 – Training & Off-chain Storage:** Each  $u \in \mathcal{U}$  trains on local data;  $a \in \mathcal{A}$  aggregates into proposal  $p_a$ .
  - 3: Upload  $p_a$  to IPFS → obtain CID $_a$ ; submit CID $_a$  on-chain.
  - 4: **Phase 3 – On-chain Consensus & Instant Update:**  $\mathcal{V}$  retrieves  $\{p_a\}$  via CIDs, verifies data availability, and runs Krum.
  - 5: Vote via PBFT. **Commit**  $w_{r+1}$  immediately upon majority.
  - 6: Record winning CID\* and voter identities on-chain. Dispense round rewards.
  - 7: **Phase 4 – Audit Window Opens:** Asynchronous challenge period begins. Nodes pin IPFS data.
- 

Validator committee retrieves these proposals, executes Krum scoring, and votes via PBFT. The winning model  $w_{r+1}$  is committed *immediately* upon achieving  $> 2/3$  committee agreement. Finally, Phase 4 initiates the asynchronous challenge window.

### 5.3 Asynchronous Audit and Challenge Mechanism

The asynchronous audit mechanism entirely decouples synchronous verification from execution. By shifting the strict correctness validation into a non-blocking background process, the system achieves maximum throughput without sacrificing long-term security.

*Challenge Trigger Logic.* Algorithm 2 details the challenge execution flow. The trigger logic relies heavily on the profound decoupling of determinism and data transparency. Because Krum is strictly deterministic, a challenger locally

---

**Algorithm 2** Asynchronous Challenge Mechanism
 

---

573  
 574     **Require:** Challenger  $ch$ , on-chain CID references, IPFS store  
 575     **Ensure:** Punishment for malicious committee actions  
 576     1:  $ch$  retrieves proposal CIDs from chain, downloads  $\{p_a\}$  from IPFS.  
 577     2:  $ch$  re-executes Krum on  $\{p_a\}$ .  
 578     3: **if** outcome mismatches committed  $w_{r+1}$  **then**  
 579         4:     $ch$  posts challenge transaction with deposit  $D_{challenge}$ .  
 580         5:    **Arbitration Triggered:** All consensus nodes verify independently.  
 581         6:    **if** malicious consensus confirmed by  $> 2/3$  of network **then**  
 582             7:      **Slash** full stake of colluding validators  $\mathcal{V}_{mal}$ .  
 583             8:      Reward Challenger  $ch$ ; distribute remainder to honest participants.  
 584             9:      // Note: Model  $w_{r+1}$  is NOT reverted.  
 585         10:    **else**  
 586             11:    Forfeit  $ch$ 's deposit  $D_{challenge}$ .  
 587

---

588  
 589  
 590     executing the algorithm on identical IPFS inputs will inevitably produce the objectively correct output. Any deviation  
 591     by the committee is thus easily recognizable.

592     When a challenger detects a discrepancy, they submit a challenge transaction enclosed with a deposit  $D_{challenge}$ . This  
 593     deposit serves two critical purposes: (1) it acts as a defensive barrier against Denial-of-Service attacks, preventing malici-  
 594     ous actors from spamming invalid challenges and exhausting network resources; and (2) it guarantees compensation  
 595     for the computational and data-transfer overhead incurred by the entire network during the arbitration process, thereby  
 596     solving the classic decentralized oversight motivation gap. Once triggered, the entire network downloads the proposals  
 597     and executes Krum. If  $> 2/3$  confirm the misbehavior, colluding committee members suffer complete stake slashing.

598  
 599  
 600  
 601     *Slashed Funds Distribution.* The distribution of confiscated funds follows a stringent incentive compatibility design.  
 602     A significant proportion of the slashed stake is awarded directly to the Challenger. This substantial bounty ensures that  
 603     monitoring the network remains economically attractive, encouraging a robust ecosystem of independent auditors. The  
 604     remaining slashed funds are distributed among the honest Update Providers who submitted valid gradients during the  
 605     compromised round, as well as the Validator nodes that actively participated in the arbitration consensus. This broad  
 606     distribution not only compensates honest participants for potential losses incurred during the attack but also cultivates  
 607     a culture of collective supervision, firmly distributing the governance power across the entire network.

608  
 609  
 610  
 611     *Endogenous Dynamic Staking Model.* A decentralized system's core parameters must derive from verifiable internal  
 612     metrics rather than external oracles, which introduce vulnerabilities (e.g., price manipulation or latency). AC-BlockDFL  
 613     anchors its penalty pricing exclusively to internal economic activity.

614  
 615     Under the rational game theory assumption, the maximum immediate gain from committee capture is bounded by  
 616     the round reward  $R_{round}$ . To build an impenetrable economic deterrent, the slashing penalty  $D_{slash}$  must dynamically  
 617     shadow  $R_{round}$ :  
 618

$$D_{slash} = \lambda \times R_{round}, \quad \lambda \gg 1 \tag{4}$$

619     By setting  $\lambda \approx 100$ , a single slashing event perfectly wipes out the equivalent of 100 rounds of honest participation.

620  
 621     Similarly, the challenge deposit  $D_{challenge}$  must simultaneously satisfy economic sustainability and accessibility. If a  
 622     challenge fails, the forfeited deposit must cover the marginal computational cost  $\epsilon$  of the  $N_{arb}$  nodes participating in  
 623  
 624     Manuscript submitted to ACM

625 arbitration (i.e.,  $D_{\text{challenge}} \geq N_{\text{arb}} \cdot \epsilon$ ). Practically,  $D_{\text{challenge}}$  can be defined as  $\alpha \times R_{\text{round}}$ . Since  $R_{\text{round}}$  scales with network  
 626 economic scale, the costs dynamically adjust to all market conditions.  
 627

628 *State Finality and No-Rollback Policy.* When arbitration confirms malicious consensus, AC-BlockDFL executes  
 629 economic slashing but deliberately *does not* revert the committed model update. This design choice addresses the  
 630 fundamental conflict between state rollback and the core blockchain principle of immutability. Reverting historical  
 631 states destroys the finality of all subsequent blocks, leaving the system highly vulnerable to *Long-Range Attacks* where  
 632 adversaries rewrite history from a deeply buried block. In a federated learning context, arbitration latency means that  
 633 by the time an early round is deemed malicious, tens or hundreds of subsequent blocks may have been appended.  
 634

635 Furthermore, rolling back the global state demands extreme coordination complexity. It requires all distributed nodes  
 636 to simultaneously revert to a historical snapshot and discard immense amounts of valid computational work, a process  
 637 that becomes exponentially more difficult as the chain grows. AC-BlockDFL instead embraces a “Forward Correction”  
 638 strategy: the severe economic penalty liquidates the attacker’s future governance influence, cutting off the attack vector  
 639 permanently. The minor mathematical deviation introduced by a single anomalous epoch is naturally digested and  
 640 rectified by the iterative self-healing property of subsequent honest training rounds.  
 641

642 **5.4 Efficiency and Overhead Analysis**  
 643 By shifting the security paradigm from threshold-based to economic-based, AC-BlockDFL successfully decouples the  
 644 stringent relationship between security guarantees and committee size, yielding substantial efficiency dividends across  
 645 communication and storage.  
 646

647 *Communication Complexity.* In standard BlockDFL deployments, suppressing the attack probability  $p_{\text{risk}}$  near zero  
 648 obligates the system to maintain a large committee. For instance, in a 100-node network with a 30% adversarial fraction,  
 649 maintaining  $p_{\text{risk}} < 0.01$  mathematically dictates  $C \geq 9$ . The resulting PBFT communication complexity acts as a rigid,  
 650 preventative premium  $O(81)$  exacted uniformly across every single round, regardless of whether the network is under  
 651 attack.  
 652

653 AC-BlockDFL neutralizes this overhead by ensuring that even if the committee is compromised, attackers cannot  
 654 profit. Consequently, the mechanism satisfies foundational security constraints with a smaller committee size  $C = 7$ ,  
 655 dropping the baseline communication complexity to  $O(49)$ —a nearly 40% reduction. The heavy network-wide PBFT  
 656 arbitration cost  $O(N^2)$  is strictly conditional. In equilibrium, because the penalty mechanism eliminates the exploit  
 657 incentive, the objective probability of an attack  $p$  approaches 0. The expected communication complexity seamlessly  
 658 converges to standard committee levels:  
 659

$$E[\text{Comm}] = (1 - p) \cdot O(C^2) + p \cdot (O(C^2) + O(N^2)) = O(C^2) + p \cdot O(N^2) \quad (5)$$

660 This establishes a highly efficient “pay-as-you-go” security model rather than a perpetual static penalty.  
 661

662 *Storage Overhead.* Recording pristine neural network gradients on an immutable ledger guarantees exponential  
 663 bloat. While IPFS cleanly mitigates on-chain storage, AC-BlockDFL uniquely incorporates a strict lifecycle management  
 664 and pinning strategy. During the challenge window, validator nodes are mathematically obligated to pin the IPFS  
 665 chunks, ensuring data availability for prospective challengers. However, the moment the challenge Time-To-Live (TTL)  
 666 window expires uninterrupted, nodes safely unpin the historical epoch’s parameters. This ephemeral storage footprint  
 667 ensures the system scales gracefully, slashing permanent on-chain storage mapping from  $O(\text{ModelSize})$  completely  
 668

Table 2. Experimental Parameters

Parameter	Value
Training rounds	$R = 300$ (baseline) / $R = 2000$ (long-term)
Validator pool size	$N = 100$
Committee size	$C = 7$
Malicious nodes	$M = 30$ (initial stake ratio 30%)
Per-round rewards	Validator: 1.0, Aggregator: 1.0, Provider: 0.05
Slashing rule	Full stake confiscation upon successful challenge

down to  $O(\text{HashSize})$ , preserving systemic decentralization without burdening node operators with unsustainable disk requirements.

## 6 Evaluation

We evaluate AC-BlockDFL through systematic experiments designed to validate its defense effectiveness against Progressive Committee Capture Attacks (PCCA). Rather than treating model accuracy as the primary metric, our evaluation focuses on whether the economic security mechanism effectively deters rational adversaries and maintains long-term governance stability. This perspective shift reflects our core design philosophy: when the defense objective transitions from “preventing attacks” to “ensuring attacks are unprofitable,” the evaluation metrics should correspondingly shift from model quality to the adversary’s economic decision space.

Our experiments adopt a worst-case analysis methodology, assuming adversaries attack whenever possible regardless of economic rationality. This design enables a critical inference: if the mechanism ensures every attack is detected and penalized under worst-case conditions, rational adversaries will preemptively conclude that expected returns are negative and abstain from attacking, allowing the system to naturally converge toward stable equilibrium.

### 6.1 Experimental Setup

We use the MNIST dataset with a standard CNN (two convolutional layers, two fully connected layers) as our federated learning testbed. Training data is distributed IID across clients—a deliberate choice since our defense operates at the consensus layer rather than the data layer. Committee composition and voting outcomes determine attack success, which are logically independent of underlying data distribution characteristics.

Table 2 summarizes the experimental configuration. The 30% initial malicious stake ratio represents a severe threat scenario approaching the theoretical tolerance limit of most Byzantine fault-tolerant systems. Under hypergeometric distribution analysis (Section 6.2), malicious nodes have approximately 2.4% probability of capturing  $\geq 5$  of 7 committee seats in any single round. While seemingly modest, this probability accumulates over hundreds to thousands of training rounds, providing ample attack opportunities for rigorous defense validation.

### 6.2 Committee Security Analysis

The probability of  $k$  malicious nodes being selected in a committee of size  $C$  from a pool  $N$  with  $M$  malicious nodes follows the hypergeometric distribution:

$$P(X = k) = \frac{\binom{M}{k} \binom{N-M}{C-k}}{\binom{N}{C}} \quad (6)$$

With  $N = 100$ ,  $M = 30$ ,  $C = 7$ , the probability of an adversarial takeover ( $k \geq 5$ ) is  $\sim 2.41\%$ .

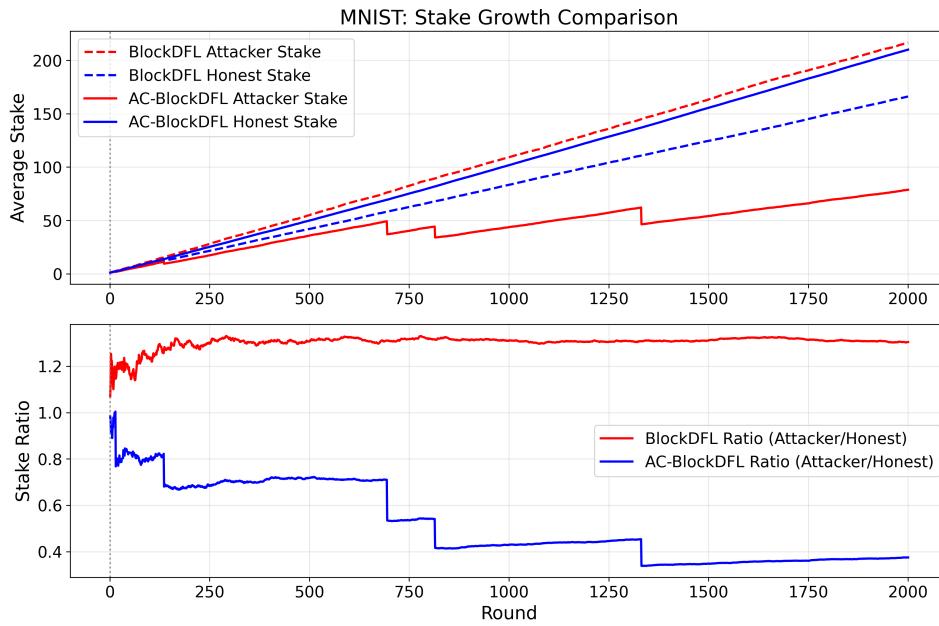


Fig. 2. Stake dynamics over 2000 rounds. BlockDFL exhibits persistent governance imbalance with malicious stake ratio stabilizing at 1.3 $\times$ , while AC-BlockDFL achieves progressive purification through five slashing events, reducing malicious stake to 0.37 $\times$  of honest nodes.

### 6.3 Long-term Governance Equilibrium

The critical question our evaluation addresses is whether short-term penalty effectiveness translates into long-term governance stability where attacks naturally cease. The 2000-round simulation provides definitive evidence.

Figure 2 reveals fundamentally divergent governance trajectories. In BlockDFL, the malicious stake ratio stabilizes around 1.3 after initial fluctuations and persists throughout the experiment. This seemingly modest advantage masks a profound governance crisis: the 1.3 $\times$  ratio translates to significantly elevated committee election probabilities, sustaining continuous attack capability across 2000 rounds. Without accountability mechanisms, adversaries reinforce their stake advantage through each successful capture, confirming the positive feedback loop predicted in Section 4.2.

AC-BlockDFL exhibits a starkly different pattern. The malicious stake ratio undergoes five distinct step-wise decreases at rounds 15, 136, 695, 815, and 1332, declining from the initial 1.0 to a final 0.37. This terminal value indicates that malicious nodes retain barely one-third the average stake of honest participants—a 1.3/0.37  $\approx$  3.5 $\times$  difference from BlockDFL representing fundamental governance reversal rather than incremental improvement.

Table 3 provides direct causal evidence for the stake trajectories. BlockDFL records 107 committee capture events (averaging one per 19 rounds), none receiving economic sanction. AC-BlockDFL records only 5 attacks, all successfully detected and penalized—a >20 $\times$  reduction stemming from two reinforcing mechanisms: slashing directly depletes the malicious stake base, reducing subsequent committee election probability; simultaneously, diminished stake ratios raise the difficulty of achieving  $\geq 5/7$  committee control even when selected.

Table 3. Attack Event Statistics (2000-round Simulation)

Metric	BlockDFL	AC-BlockDFL
Total attacks	107	5
Strategic starvation	18	2
Full-stack poisoning	89	3
Detected & slashed	0	5 (100%)
Final stake ratio	1.30	0.37

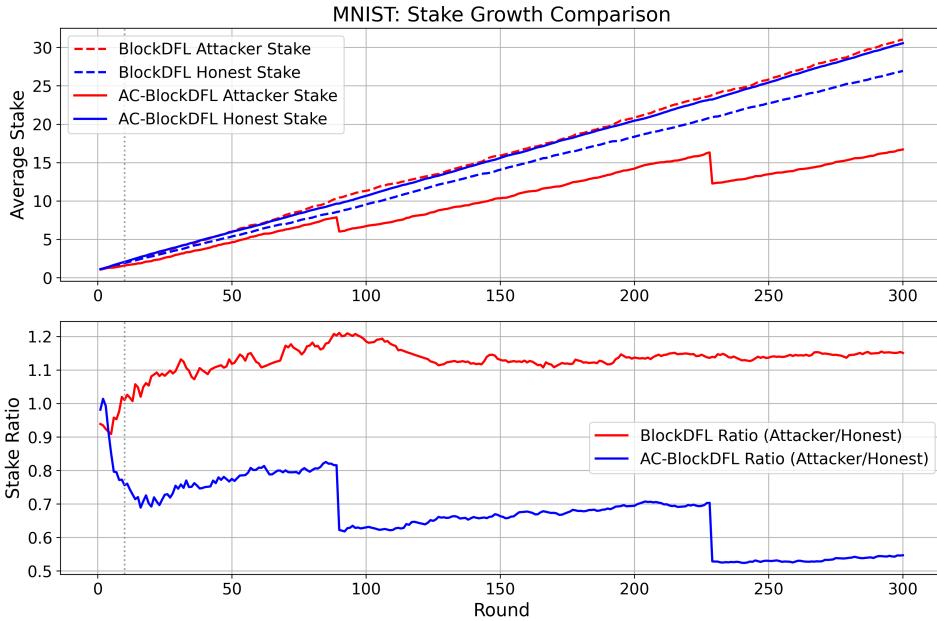


Fig. 3. Stake evolution comparison (300-round baseline). AC-BlockDFL demonstrates immediate stake ratio drops upon each slashing event, while BlockDFL shows continuous malicious stake accumulation.

The increasing intervals between slashing events provide key evidence of convergence toward equilibrium. Specifically: 121 rounds between events 1–2, 559 rounds between 2–3, 120 rounds between 3–4, 517 rounds between 4–5, and 668 rounds of silence following the fifth event through experiment termination. This pattern is not statistical noise but a mathematical consequence of stake depletion: as malicious stake fraction decreases from 30% toward 20%, the single-round probability of achieving committee control drops from ~2.4% to <0.5%, directly manifesting as attack window rarefaction. The 668-round silent period following the final slashing confirms the system has converged to a state where attacks become structurally improbable.

#### 6.4 Immediate Mechanism Response

The 300-round baseline experiment provides a controlled window for examining the immediate impact of individual slashing events on governance structure.

Manuscript submitted to ACM

833 Figure 3 shows the early-stage stake trajectories. In BlockDFL, 10 committee capture events occur over 300 rounds (4  
 834 strategic starvation, 6 full-stack poisoning), all unpunished, enabling the malicious stake ratio to climb steadily from 1.0  
 835 toward 1.15. AC-BlockDFL records only 2 attacks at rounds 90 and 229, both detected and slashed with 100% accuracy.  
 836

837 The first slashing event illustrates the mechanism’s precision. By round 90, malicious nodes had accumulated a 1.25  
 838 stake ratio through honest participation, translating to elevated committee selection probability. When 5 malicious  
 839 nodes achieved committee control and executed full-stack poisoning, a challenger detected the deviation by locally  
 840 re-executing Krum aggregation and submitted a challenge transaction. Upon arbitration confirmation, the smart contract  
 841 automatically confiscated the full stakes of all 5 colluding members. The economic impact was immediate and severe:  
 842 the malicious stake ratio plummeted from 1.25 to 0.62—a single event reversing the adversary’s 25% lead into a 38%  
 843 deficit.

844 This magnitude of impact warrants careful interpretation. The slashed nodes lost not merely the current round’s  
 845 potential gains (bounded by ~7.0 reward units) but their entire accumulated stake from 89 rounds of honest participation.  
 846 More critically, stake-zeroed nodes are effectively excluded from future high-reward role elections, constituting  
 847 “permanent governance exclusion” that degrades long-term attack capability beyond the immediate economic penalty.  
 848

849 The second slashing at round 229 reduced the stake ratio from 0.70 to 0.52. The 139-round interval between attacks  
 850 (versus BlockDFL’s average of 30 rounds) directly reflects the first slashing’s suppressive effect on attack opportunity  
 851 windows.

## 852 6.5 Service Quality Under Security Guarantees

853 A critical concern is whether security guarantees impose unacceptable performance costs. We evaluate both system  
 854 availability and model convergence quality.

855 *System Availability.* We define minimum unavailability rate as the fraction of rounds where model performance is  
 856 significantly degraded due to full-stack poisoning attacks. Each attack requires approximately 5–25 rounds for federated  
 857 learning’s self-healing mechanism to restore accuracy. Using the conservative 5-round estimate, BlockDFL’s 89 full-stack  
 858 attacks yield a minimum unavailability rate of  $89 \times 5 / 2000 = 22.3\%$ . AC-BlockDFL achieves  $3 \times 5 / 2000 = 0.75\%$ —a >96%  
 859 improvement attributable entirely to attack frequency suppression rather than enhanced per-attack resilience.

860 *Model Convergence and Training Stability.* Figure 4 compares accuracy trajectories over the 300-round baseline.  
 861 BlockDFL’s accuracy curve exhibits pronounced sawtooth patterns, with each severe drop precisely corresponding  
 862 to a full-stack poisoning attack. Notably, the earliest threats in BlockDFL did not manifest as accuracy drops. The  
 863 initial attack at round 69 was a stealthy “strategic starvation” maneuver. By rejecting honest proposals and approving  
 864 suboptimal ones that favored malicious providers, the adversary manipulated the economic flow to accelerate stake  
 865 accumulation. From an accuracy monitoring perspective, strategic starvation leaves almost no anomalies, confirming  
 866 our argument (Section 4.2) that relying solely on model quality metrics creates a fundamental security blind spot. As  
 867 malicious nodes built their stake advantage through starvation, attack frequency accelerated (averaging one per 19  
 868 rounds), eventually escalating to full-stack poisoning. Although federated learning’s resilience allows gradual recovery,  
 869 this continuous “deviation and correction” cycle severely degrades computational efficiency.

870 Conversely, AC-BlockDFL’s accuracy curve demonstrates fundamentally different dynamic stability. During the  
 871 300-round period, it suffered only 2 full-stack attacks. The extreme case at round 90 provides an excellent benchmark: the  
 872 adversary executed a label-flipping attack that crashed accuracy from normal levels down to 9.5% (near random-guess  
 873 baseline for MNIST’s 10-class task). However, the system showcased remarkable self-healing capability, recovering to  
 874

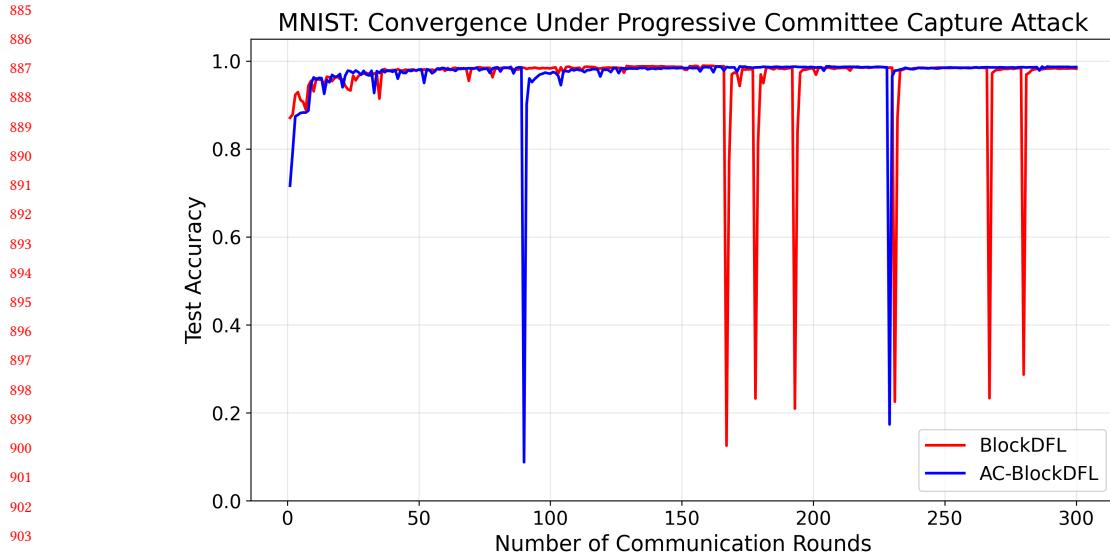


Fig. 4. Model accuracy convergence comparison. AC-BlockDFL exhibits smoother training dynamics with fewer disruption-recovery cycles.

Table 4. Communication Complexity Comparison

Scheme	Complexity	Overhead (MB/round)
Full BFT	$O(N^2)$	25.4
BlockDFL	$O(C^2)$	4.2
AC-BlockDFL	$O(pN^2 + C^2)$	4.3

pre-attack levels within  $\approx 20$  rounds. More importantly, as training progressed and the slashing mechanism purged malicious stake, the system’s resilience improved. In late-stage training, the robust parameter structure established during uninterrupted honest epochs reduced the recovery time for equivalent disturbances from 20 rounds down to just  $\approx 5$  rounds.

Both architectures reach similar final accuracies (98.26% vs. 98.63%), confirming that the “no-rollback” design philosophy (Section 5.3) is practically sound: federated learning’s iterative nature inherently digests occasional deviations, obviating the immense coordination overhead of state rollbacks. Yet, AC-BlockDFL’s near-interference-free environment ensures computing resources are efficiently translated into optimization gains rather than wasted on repairing attack damage—a critical advantage for resource-constrained edge deployments.

*Communication Efficiency.* As analyzed in Section 6.5 and summarized in Table 4, AC-BlockDFL achieves  $O(C^2)$  communication complexity under equilibrium conditions where the challenge trigger probability  $p \rightarrow 0$ . Compared to approaches requiring equivalent security guarantees through full replication, this represents approximately 40% reduction in per-round communication overhead while maintaining the same Byzantine tolerance threshold.

## 937 6.6 Summary

938 Our evaluation validates AC-BlockDFL’s defense effectiveness through three complementary lenses. At the micro level,  
 939 each malicious committee decision triggers immediate detection and slashing with 100% accuracy. At the macro level,  
 940 five slashing events progressively reduce the malicious stake ratio from 1.0 to 0.37, with increasing inter-event intervals  
 941 and a terminal 668-round silent period confirming convergence to attack-free equilibrium. Service quality analysis  
 942 demonstrates that these security guarantees impose minimal performance cost: unavailability rate drops from 22.3% to  
 943 0.75%, while model convergence remains uncompromised. These results complete the inference chain: worst-case testing  
 944 proves all attacks are detected; rational adversaries therefore anticipate penalties and abstain; the system operates at  
 945 designed efficiency under the resulting equilibrium.

## 949 7 Conclusion

950 This paper identifies and formalizes the Progressive Committee Capture Attack (PCCA), demonstrating how rational  
 951 adversaries can systematically compromise committee-based blockchain federated learning systems through strategic  
 952 stake accumulation. Our long-horizon simulations confirm that conventional committee architectures exhibit stake  
 953 ossification and governance capture under sustained attack.

954 To address this threat, we propose AC-BlockDFL, an audit-driven committee architecture that decouples security  
 955 guarantees from committee size. The key insight underlying our design is a paradigm shift from *threshold security*—  
 956 which seeks to minimize the probability of committee compromise—to *economic security*—which ensures that even  
 957 successful compromise yields negative expected utility for rational adversaries. Through asynchronous auditing and the  
 958 internal slashing protocol, AC-BlockDFL achieves progressive purification of malicious participants while maintaining  
 959 the efficiency benefits of small committees.

960 Our experimental results validate three principal contributions: (1) formal threat modeling of PCCA with empirical  
 961 verification of its feasibility; (2) demonstration that slashing mechanisms effectively break the positive feedback loop of  
 962 malicious stake accumulation, internalizing the externalities of adversarial behavior; and (3) evidence that shifting from  
 963 preventive to reactive security breaks the tight coupling between security guarantees and communication overhead,  
 964 enabling practical deployment in resource-constrained edge computing scenarios.

## 965 References

- [1] Anonymous. 2024. FedChain: Secure and Efficient Federated Learning via Blockchain. *Under Review* (2024).
- [2] Eugene Bagdasaryan, Andreas Veit, Yiqing Hua, Deborah Estrin, and Vitaly Shmatikov. 2020. How to backdoor federated learning. In *International conference on artificial intelligence and statistics*. PMLR, 2938–2948.
- [3] Peva Blanchard et al. 2017. Machine Learning with Adversaries: Byzantine Tolerant Gradient Descent. In *Advances in Neural Information Processing Systems (NeurIPS)*.
- [4] Miguel Castro and Barbara Liskov. 1999. Practical Byzantine Fault Tolerance. In *Proceedings of the 14th USENIX Conference on Operating Systems Design and Implementation (OSDI)*.
- [5] Bing-Jyue Chen, Suppakit Waiwitlikhit, Ion Stoica, and Daniel Kang. 2024. Zkml: An optimizing system for ml inference in zero-knowledge proofs. In *Proceedings of the Nineteenth European Conference on Computer Systems*. 560–574.
- [6] Jonathan Chiu and Thorsten Koepll. 2018. The incentives of blockchain and the optimal design of cryptocurrencies. *Review of Financial Studies* (2018).
- [7] C. Conway, C. So, X. Yu, and K. Wong. 2024. opML: Optimistic machine learning on blockchain. *arXiv preprint arXiv:2401.17555* (2024).
- [8] M. Elmahallawy et al. 2025. Decentralized Federated Learning for Satellite Networks. *arXiv preprint arXiv:2501.xxxxx* (2025).
- [9] EZKL. 2024. Benchmarking ZKML frameworks. <https://blog.ezkl.xyz/> EZKL Blog.
- [10] M. Fang, X. Liu, N. Liao, and N. Z. Gong. 2020. Local Model Poisoning Attacks to Byzantine-Robust Federated Learning. In *Proc. 29th USENIX Security Symp. (USENIX Security)*. Boston, MA, USA, 1623–1640.

- 989 [11] Jonas Geiping, Hartmut Bauermeister, Hannah Dröge, and Michael Moeller. 2020. Inverting Gradients - How easy is it to break privacy in federated  
990 learning?. In *Advances in Neural Information Processing Systems (NeurIPS)*, Vol. 33. 16937–16947.
- 991 [12] Yossi Gilad et al. 2017. Algorand: Scaling Byzantine Agreements for Cryptocurrencies. In *Proceedings of the 26th Symposium on Operating Systems  
992 Principles (SOSP)*.
- 993 [13] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles,  
994 Graham Cormode, Rachel Cummings, et al. 2021. Advances and open problems in federated learning. *Foundations and trends® in machine learning*  
995 14, 1–2 (2021), 1–210.
- 996 [14] Leslie Lamport, Robert Shostak, and Marshall Pease. 2019. The Byzantine generals problem. In *Concurrency: the works of leslie lamport*. 203–226.
- 997 [15] D. Li et al. 2021. A Blockchain-Based Federated Learning Framework with Committee Consensus. *IEEE Network* (2021).
- 998 [16] Y. Liu et al. 2021. Blockchain-Enabled Federated Learning for Vehicular Networks. *IEEE Transactions on Vehicular Technology* (2021).
- 999 [17] Y. Lu et al. 2020. Blockchain-enabled federated learning for industrial IoT. *IEEE Transactions on Industrial Informatics* (2020).
- 1000 [18] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep  
1001 networks from decentralized data. In *Artificial intelligence and statistics*. PMLR, 1273–1282.
- 1002 [19] Yinbin Miao, Ziteng Liu, Hongwei Li, Kim-Kwang Raymond Choo, and Robert H Deng. 2022. Privacy-preserving Byzantine-robust federated  
1003 learning via blockchain systems. *IEEE Transactions on Information Forensics and Security* 17 (2022), 2848–2861.
- 1004 [20] H. Nguyen et al. 2024. FedBlock: A Blockchain-Based Federated Learning Framework with Adaptive Committee Selection. *IEEE Transactions on  
Parallel and Distributed Systems* (2024).
- 1005 [21] Y. E. Octavian and S.-G. Lee. 2023. Blockchain-Based Federated Learning System: A Survey on Design Choices. *Sensors* (2023).
- 1006 [22] Optimism Foundation. 2024. Rollup protocol overview. <https://docs.optimism.io>. Accessed: 2026-02-02.
- 1007 [23] ORAProtocol. 2024. opML documentation. <https://docs.ora.io>. Accessed: 2026-02-02.
- 1008 [24] Shiva Raj Pokhrel. 2021. Blockchain brings trust to collaborative drones and LEO satellites: An intelligent decentralized learning in the space. *IEEE  
Sensors Journal* 21, 14 (2021), 15731–15741.
- 1009 [25] S. R. Pokhrel and J. Choi. 2020. Autonomous Vehicles in 5G and Beyond: A Blockchain-Based Federated Learning Approach. *IEEE Transactions on  
Intelligent Transportation Systems* (2020).
- 1010 [26] Jiaming Qin et al. 2024. BlockDFL: A Blockchain-based Fully Decentralized Peer-to-Peer Federated Learning Framework. In *Proceedings of the the  
1011 IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*.
- 1012 [27] Y. Qu et al. 2020. Decentralized Federated Learning: A Survey. *IEEE Communications Surveys & Tutorials* (2020).
- 1013 [28] Y. Ren et al. 2024. A scalable blockchain-enabled federated learning architecture for edge computing. *PLOS ONE* (2024).
- 1014 [29] Muhammad Shayan et al. 2021. Biscotti: A Blockchain System for Private and Secure Federated Learning. In *IEEE Transactions on Parallel and  
Distributed Systems*.
- 1015 [30] Z. Wang, W. Liang, J. Wang, K. Yu, X. Du, and M. Guizani. 2024. A Secure and Efficient Federated Averaging based on Zero-Knowledge Proofs.  
1016 *Electronics* 13, 1 (Jan. 2024), 195. doi:10.3390/electronics13010195
- 1017 [31] J. Weng et al. 2021. DeepChain: Auditable and Privacy-Preserving Deep Learning with Blockchain. *IEEE Transactions on Dependable and Secure  
1018 Computing* (2021).
- 1019 [32] Y. Wu et al. 2024. A Sharded Blockchain-Based Secure Federated Learning Framework for LEO Satellite Networks. *IEEE Transactions on Network  
1020 and Service Management* (2024).
- 1021 [33] Z. Xing, Zijian Zhang, Ziang Zhang, Z. Li, M. Li, J. Liu, et al. 2023. Zero-Knowledge Proof-based Verifiable Decentralized Machine Learning in  
1022 Communication Network: A Comprehensive Survey. *arXiv preprint arXiv:2312.00000* (2023).
- 1023 [34] D. Yin et al. 2018. Byzantine-Robust Distributed Learning: Towards Optimal Statistical Rates. In *International Conference on Machine Learning  
(ICML)*.
- 1024 [35] Mahdi Zamani, Mahnush Movahedi, and Mariana Raykova. 2018. Rapidchain: Scaling blockchain via full sharding. In *Proceedings of the 2018 ACM  
1025 SIGSAC conference on computer and communications security*. 931–948.
- 1026 [36] Ligeng Zhu, Zhijian Liu, and Song Han. 2019. Deep leakage from gradients. *Advances in neural information processing systems* 32 (2019).
- 1027 [37] Yizheng Zhu, Yuncheng Wu, Zhaojing Luo, Beng Chin Ooi, and Xiaokui Xiao. 2023. Secure and verifiable data collaboration with low-cost  
1028 zero-knowledge proofs. *arXiv preprint arXiv:2311.15310* (2023).
- 1029
- 1030
- 1031
- 1032
- 1033
- 1034
- 1035
- 1036
- 1037
- 1038
- 1039
- 1040 Manuscript submitted to ACM