

Scalable and Secure Framework for Federated Learning in IoV applying Committee-based Chained Hotstuff

Juyoung Yoon
*Division of Artificial Intelligence
 and Software*
Ewha Womans University
 Seoul, 03760, Republic of Korea
 yjy011123@ewha.ac.kr

Gabin Heo
*Division of Artificial Intelligence
 and Software*
Ewha Womans University
 Seoul, 03760, Republic of Korea
 gjrkqls@ewhain.net

Inshil Doh
Department of Cyber Security
Ewha Womans University
 Seoul, 03760, Republic of Korea
 isdoh1@ewha.ac.kr

Abstract—Applying Federated Learning (FL) in the Internet of Vehicles (IoV) presents a critical trade-off between privacy, security, and performance, which conventional blockchains fail to address due to prohibitive latency and overhead. To overcome these limitations, we propose a novel framework featuring a stack-based verification engine for deterministic validation, an on-chain management system for conditional privacy, and a Committee-based Chained HotStuff (C-CHS) consensus algorithm. The C-CHS protocol fundamentally resolves the leader bottleneck by enabling parallel block preparation. Our evaluation demonstrates substantial performance gains: C-CHS increases throughput by up to 92% (vs. CHS) and 873% (vs. PBFT), while reducing latency by up to 52% and 90%, respectively. These results validate our framework as a secure, scalable, and practical solution for deploying FL in large-scale IoV environments.

Index Terms—Blockchain, IoV, Conditional Privacy, Hotstuff, Federated Learning

I. INTRODUCTION

The Internet of Vehicles (IoV) environment, where vehicles, Roadside Units (RSUs), and the cloud are interconnected, enables innovative services such as autonomous driving model updates and real-time traffic optimization. However, this dynamic setting faces persistent challenges in ensuring security, privacy, and scalability. Especially, when applying **Federated Learning (FL)** to protect data privacy, complex issues arise, including the need to guarantee the integrity of model updates submitted by individual vehicles and to ensure accountability for tracing malicious acts committed under the cover of anonymity [1].

These requirements can only be met in a trust-distributed, multi-agent environment where all participants record and verify data according to the same set of rules. To achieve this, the adoption of blockchain technology, which provides strong integrity and auditability, is essential. **Blockchain** can serve as the trust layer for IoV through its immutable, distributed ledger [2]. However, considering the high mobility and real-time requirements of IoV, the overhead and latency issues inherent in the consensus processes of conventional blockchains are critical challenges that must be addressed. Furthermore,

inadequate on-chain privacy protection can lead to privacy breaches through traceability.

To address these challenges, this paper proposes a novel IoV-FL framework that organically combines **stack-based verification**, **on-chain pseudonym management**, and a **committee-based Chained HotStuff consensus algorithm**. In the proposed framework, all transactions—such as vehicle registration, model submission, and pseudonym changes—are deterministically verified by all nodes using lightweight stack scripts. Additionally, it ensures Conditional Privacy by binding a vehicle’s public key and its pseudonym (HID) on-chain and establishing a restricted link that allows a trusted authority to trace it only when necessary. Finally, during the consensus process, the leader prepares the next block in parallel while committee nodes aggregate signatures, thereby minimizing consensus latency and maximizing the overall system throughput.

The key contributions of this paper are as follows:

- 1) **Stack-Scripted RSU Verification with On-Chain HID-PK Binding and Conditional Traceability:** We designed a system where vehicle registration and model submission procedures are standardized through stack-based scripts, enabling all RSUs to perform rapid verification under a uniform set of rules. Also, we achieve a balance between privacy and accountability through the on-chain binding of pseudonyms (HIDs) and public keys, coupled with a restricted traceability link.
- 2) **Committee-Aggregated Chained HotStuff with Parallel Block Preparation:** To resolve the leader bottleneck issue in the original Chained HotStuff, we introduced a committee-based signature aggregation and parallel block preparation architecture. This approach reduces latency in the consensus process, significantly enhancing the throughput and stability of the entire network.

The remainder of this paper is organized as follows. Section II discusses related work. Section III presents the system model and design goals. Section IV details the stack-based verification and on-chain pseudonym management mechanisms.

Section V proposes the committee-based Chained HotStuff algorithm. Section VI analyzes the experimental environment and evaluation results. Finally, Section VII concludes the paper.

II. RELATED WORK

A. Privacy-Preserving and Pseudonym Management in IoV

In the Internet of Vehicles (IoV), Conditional Privacy-Preserving Authentication (CPPA), which conceals a vehicle's real identity while allowing it to be traced only when necessary, is crucial. Cheng et al. [3] addressed cross-domain authentication in 6G-IoV by proposing pairing-free Identity-Based Inspection for batch verification and Blockchain-Assisted Pseudonym Management. To reduce the overhead of pseudonym lifecycle management, they employed a Dynamic Sparse Merkle Tree. Xiong et al. [4] combined PUF with blockchain, enabling vehicles to generate dynamic pseudonyms with one-time keys, thus eliminating the need for pre-storage. Verification is then performed efficiently using Simplified Payment Verification. Zhang et al. [5] implemented a blockchain-based trust management system where reputation scores are recorded on-chain. They utilized a hybrid PoW and PoS consensus to prioritize updates from vehicles with significant reputation changes. Nevertheless, the following challenges remain. Some schemes still presume the use of multiple pseudonyms, which creates significant storage and management overhead for both vehicles and the system. And procedures for pseudonym replacement and verification can be complex, and frequent round-trip interactions with the blockchain can lead to increased latency.

B. Scalable BFT Consensus Algorithms

In IoV, applications of PBFT algorithm have been widely used in practical and prototype stages. However, their scalability is limited because the three-phase broadcast protocol increases communication complexity to $O(n^2)$, and the view change process is cumbersome. To mitigate these limitations, Yin et al. [6]'s HotStuff achieved linear view change and responsiveness through threshold signatures and a 3-chain pipeline. However, it was noted that forks during the pipeline process could potentially increase latency. Jalalzai et al. [7]'s Fast-HotStuff prevents forks by requiring a "proof of extending the latest block," further reducing latency with a two-round process in the normal path. In the trend of reducing the leader bottleneck, Zhu et al. [8]'s P-HotStuff shortened waiting periods by parallelizing the Broadcast and Agreement phases. Alqahtani et al. [9]'s BigBFT mitigated the single-leader bottleneck with multi-leader support, vote piggybacking, and inter-round pipelining. In summary, while scalability has improved, issues such as idle time while waiting for a Quorum Certificate (QC), task concentration on a single leader, and view change overhead still remain. This paper aims to propose a solution to reduce these problems.

III. SYSTEM MODEL AND DESIGN GOALS

A. System Overview

This paper assumes a three-tier IoV architecture consisting of vehicles, Roadside Units (RSUs), and a central server. The RSU network maintains the distributed ledger and deterministically verifies the validity of data submitted by vehicles using stack-based scripts. Privacy is ensured through the on-chain binding of public keys and pseudonyms (HIDs) to provide conditional privacy. For consensus, we adopt Chained HotStuff but mitigate the leader bottleneck by employing a committee to collect and aggregate votes. The overall architecture is depicted in Figure 1.

The roles of each component are as follows:

- **Vehicle:** Through an onboard HSM, a vehicle generates a private-public key pair and creates a pseudonym (HID) based on them for communication with RSUs. It submits data such as local model parameters from its participation in Federated Learning (FL) to a nearby RSU.
- **RSU (Roadside Unit):** RSUs are the consensus nodes participating in the blockchain network. They deterministically verify requests received from vehicles using stack scripts and include valid transactions in blocks. During the consensus process, each RSU performs one of the following three roles:
 - *Leader:* Proposes new blocks.
 - *Committee:* Collects and aggregates votes from replicas to generate a Quorum Certificate (QC).
 - *Replica:* Verifies proposed blocks and casts votes.
- **Server:** This is a trusted authority responsible for managing the Federated Learning epochs and the system's audit keys. It approves vehicle pseudonym change requests and holds the authority to trace a vehicle's past pseudonyms—only under legitimate procedures—through an encrypted pointer recorded on-chain.

B. Threat Model and Design Goals

This study assumes the following threat model. Given a network with n total RSU nodes and f malicious RSUs capable of Byzantine faults, the system ensures Byzantine Fault Tolerance (BFT) by satisfying the condition $n \geq 3f + 1$. Byzantine RSUs are assumed to be able to attempt message delay, loss, and replay attacks, or cast malicious votes. However, they cannot break the underlying cryptographic computations.

Within this environment, our framework aims to achieve the following design goals:

- 1) **Unlinkability and Conditional Traceability:** The real identity of a vehicle must not be exposed, and under normal circumstances, linking different pseudonyms should be infeasible. However, in response to a legitimate request, past pseudonyms must be traceable, but only through the intervention of the server.
- 2) **Data Integrity and Anti-Forgery:** All messages and model parameters transmitted by a vehicle must be authenticated via digital signatures. Furthermore, forgery

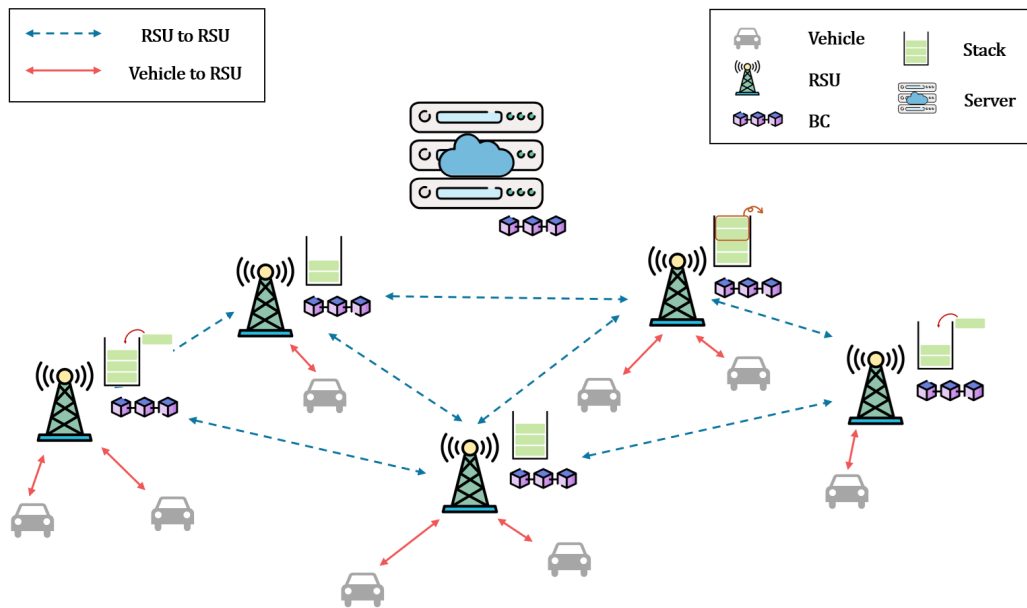


Fig. 1. The overall system architecture of the proposed IoV framework.

and tampering must be fundamentally prevented through stack-based policy verification.

- 3) **Scalability and Efficiency:** The consensus process involves only RSU nodes, offloading the burden from vehicles. Additionally, by delegating vote aggregation and QC generation to a committee, the framework must alleviate the leader bottleneck, thereby improving the overall system throughput and latency stability.

IV. STACK-BASED PROTOCOL DESIGN

For the RSU's verification mechanism, this study adopts a stack-based approach to overcome the limitations of conventional local code implementations. While a stack-based execution environment is intentionally designed to be Turing-incomplete, which imposes clear limits on its expressiveness and flexibility, this "limitation" is instrumental in enhancing security and reliability in a large-scale, distributed environment where RSUs from various manufacturers coexist.

The key advantages are as follows:

- **Enhanced Security and Stability:** The stack execution environment operates as a functionally limited sandbox, reducing the attack surface on the RSU's core system. Furthermore, its loop-free design mitigates the risk of resource exhaustion attacks, thereby contributing to system stability.
- **Consistency through Deterministic Execution:** A stack script is deterministic, meaning it produces the same output regardless of the execution environment. This provides a foundation for all nodes to operate under a unified set of rules, even in a heterogeneous environment with multi-vendor RSUs.
- **Increased Maintenance Efficiency:** When verification policies change, lightweight scripts can be dynamically

deployed instead of requiring a full firmware update for each RSU. This alleviates the overall system maintenance burden and enables a more agile response to policy changes.

The primary stack opcodes used in this framework are as follows:

- **OP_DUP, OP_SWAP:** Duplicates the top item on the stack or swaps the positions of the top two items.
- **OP_CAT, OP_HASH:** Concatenates two data items or applies a cryptographic hash function.
- **PUSH:** Pushes external data onto the stack.
- **OP_EQUALVERIFY:** Compares the top two stack items for equality and immediately fails the verification if they are not identical.
- **OP_CHECKSIG:** Verifies the validity of data using a given public key and signature.
- **OP_CHECKCERTVALID:** A custom opcode that verifies the signature and validity period of a manufacturer's certificate.
- **OP_VERIFYBINDCERT:** A custom opcode that verifies whether a certificate issued by the committee correctly maps a specific pseudonym to a public key.

A. Vehicle Registration Protocol

Vehicle registration is the process of securely binding a vehicle's identity (public key) and pseudonym (HID) on the blockchain when it first joins the IoV network.

- 1) **Key and Pseudonym Generation:** The vehicle generates a private-public key pair via its HSM and receives a certificate from its manufacturer. It then generates a cryptographically secure random salt. The pseudonym is

calculated by hashing the concatenated public key and salt value.

$$\text{HID} = H(pk_{dev} || \text{salt})$$

- 2) **Registration Request:** The vehicle creates a RegisterReq message with the following structure and transmits it to a nearby RSU.

RegisterReq = {HID, pk_dev, timestamp, sig_dev, cert_manu, salt}

- 3) **Committee Verification:** Upon receiving the request, the RSU broadcasts it to the committee members. The committee RSUs sequentially perform three verifications.

- **Pseudonym Verification:** Re-calculates the pseudonym with the given public key and salt to confirm it matches the HID in the message.
- **Manufacturer Certificate Verification:** Uses OP_CHECKCERTVALID to validate the manufacturer's certificate and confirm that the public key within it matches the vehicle's public key.
- **Ownership Signature Verification:** Uses OP_CHECKSIG to verify the vehicle's signature, confirming that the request was generated by the owner of the corresponding private key.

- 4) **Consensus and Registration:** If a sufficient number of committee RSUs successfully verify the request, the committee generates a BindCert and sends it to the vehicle.

BindCert = {hid, pk_dev, timestamp, sig_committee}

Subsequently, a RegisterTx transaction is recorded on-chain as follows:

RegisterTx = {type:register, pk_dev, hid, sig_committee, timestamp}

B. Model Parameter Submission Protocol

Vehicles participating in Federated Learning must periodically submit their trained model parameters. As this occurs frequently, a nearby RSU performs a rapid verification without requiring committee consensus.

- 1) **Submission Message Generation:** The vehicle creates and sends a SubmitMsg to an RSU.

SubmitMsg = {submit, epoch_num, hid, w, pk_dev, BindCert, timestamp, sig}

- 2) **RSU Verification:** The RSU sequentially performs two verifications using the following stack script:

- **Certificate Verification:** Uses OP_VERIFYBINCERT to confirm, via the committee's signature, that the submitted BindCert correctly links the pseudonym and the public key.
- **Ownership Signature Verification:** Uses OP_CHECKSIG to verify the submission signature and confirm message ownership.

- 3) **On-chain Recording:** Once verified, the message is converted by the RSU into a ModelTx transaction and recorded on the blockchain.

ModelTx = {model, epoch_num, hid, w}

C. Pseudonym Change Protocol

To protect vehicle privacy, pseudonyms are changed at the end of each epoch, mediated by the server.

- 1) **Change Request:** The vehicle sends a ChangeReq message to the server.

ChangeReq = {old_hid, new_hid, BindCert, pub_dev, timestamp, sig_dev}

- 2) **Server Verification:** The server verifies the following:

- It validates the BindCert to confirm that the old_hid belongs to the vehicle.
- It verifies the request signature to confirm the request is from the vehicle itself.
- It checks that the new_hid is unique and has not been used before.

- 3) **On-chain Recording:** Upon successful verification, the server creates and records a ChangeTx transaction on the blockchain.

ChangeTx = {changehid, epoch_num, new_hid, enc_old, ta_sig}

Here, enc_old is the previous pseudonym encrypted with the server's public key. This prevents external parties from linking the two pseudonyms, but allows the server to decrypt and trace the history if necessary.

V. DISTRIBUTED LEDGER AND CONSENSUS PROTOCOL

A. Ledger Structure Optimized for Federated Learning

The distributed ledger in this framework is synchronized with the temporal structure of Federated Learning (FL). The FL process operates in **Epochs** managed by the server, with each Epoch consisting of multiple **Rounds**. This hierarchical structure is illustrated in Figure 2. To align with this structure, the blockchain generates two types of blocks:

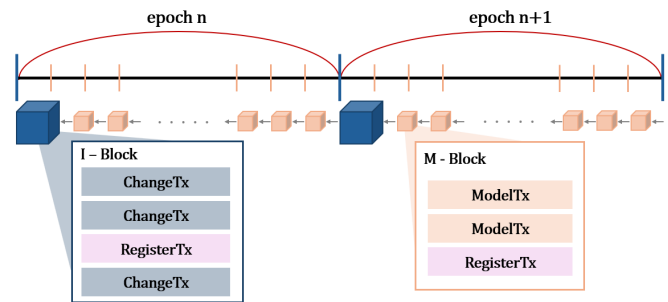


Fig. 2. The block structure of each epochs and rounds

- **I-Block (Identity Block):** This block is created at the beginning of each Epoch and exclusively contains registration and change transactions for the pseudonyms to be used during that Epoch. This separation of identity-related records facilitates easier auditing and traceability.
- **M-Block (Model Block):** This block is generated in each Round to record the model parameter transactions submitted by the vehicles.

Furthermore, at the start of each Epoch, the server evaluates the online status and performance of RSU nodes from the preceding Epoch. It then randomly elects a new leader and committee from the top 30% of active, stably connected nodes. This process prevents the centralization of authority in specific nodes and maintains the overall network stability.

B. Limitations of Chained HotStuff

HotStuff is a BFT-family consensus algorithm that achieves consensus by aggregating a threshold of signatures from a multitude of nodes to create a Quorum Certificate (QC), delivering high performance through optimistic responsiveness. Chained HotStuff enhances this by maximizing throughput in a pipelined fashion, allowing the consensus for the next block to begin before the current block's consensus is fully complete.

However, in the original Chained HotStuff, the leader must directly collect a threshold of signatures from all replicas to create the QC. This process forces the leader to wait for signatures to arrive, resulting in **idle time** and creating a leader bottleneck. Moreover, the practice of rotating the leader for every block can introduce frequent synchronization overhead, which may be unsuitable for dynamic environments like IoV.

C. Committee-based Chained HotStuff

To overcome these limitations, this paper introduces a **Committee** and proposes a Committee-based Chained HotStuff, which fixes the leader and committee for the duration of an Epoch. The proposed consensus protocol, illustrated in Figure 3, operates as a four-stage pipeline:

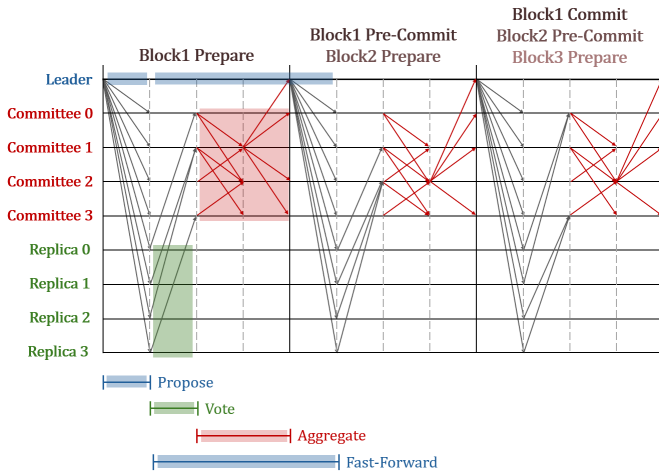


Fig. 3. The proposed Committee-based Chained HotStuff.

- 1) **Propose:** The leader proposes a block for the current round to all RSU nodes. Concurrently, the leader begins preparing the transactions to be included in the next round's block.
- 2) **Vote:** Upon receiving the proposed block, each replica validates it. If valid, the replica generates a threshold signature and sends it to a random node within the committee.

- 3) **Aggregate:** Committee nodes exchange partial signatures among themselves using a gossip protocol. The first committee node to collect a sufficient number of valid signatures aggregates them to create a QC, which is then sent to the leader and other committee members. During this phase, the leader is not idle but continues the preparation of the next block.
- 4) **Fast-forward:** The committee node that first reaches the quorum threshold generates the QC and distributes it to the leader and other committee members. Upon receiving the QC, the leader immediately verifies it, sets it as the parent QC, finalizes the pre-prepared next block, and instantly proposes it to the network.

Through this architecture, the burden of signature collection and QC generation is delegated to the committee, allowing the leader to minimize idle time and focus solely on block proposals. Consequently, consensus latency is significantly reduced, and the overall system's throughput and stability are enhanced.

VI. EVALUATION

Experimental Setup

This paper provides a quantitative comparison of PBFT, Chained HotStuff (CHS), and our proposed Committee-based Chained HotStuff (C-CHS). All experiments were conducted on a machine equipped with an Intel Core i7-10700 CPU and an NVIDIA GeForce GTX 1660 GPU, under an identical workload and network model to ensure a fair comparison. All three algorithms use the same set of nodes, transaction generation parameters, and consensus parameters. The number of nodes (n) was varied as $\{25, 50, 75, 100\}$, and throughput and latency were measured over multiple rounds for each configuration. The transaction payload was fixed at an average of 12 KB ($\pm 10\%$), and the actual payload was included in proposal messages to reflect propagation delays. The network was standardized with an RTT of approximately 10 ms (1 ms jitter standard deviation) and a bandwidth of 1 Gbps. The committee size for C-CHS was set to 10% of the total number of nodes.

A. Throughput vs. Number of Nodes

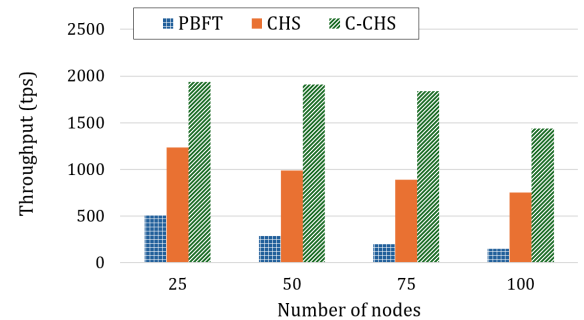


Fig. 4. Throughput comparison across different numbers of nodes.

As illustrated in Figure 4, C-CHS consistently achieves the highest throughput, starting at 1939.4 TPS ($n = 25$) and gradually decreasing by only 26% to 1435.1 TPS ($n = 100$). This represents a significant improvement of up to 92% over CHS (749.3 TPS) and 873% over PBFT (147.5 TPS) at $n = 100$. This advantage can be attributed to the reduction of pipeline stalls by overlapping the leader's waiting period with the committee's aggregation work. In contrast, PBFT's performance degrades sharply due to its cumulative broadcast overhead.

B. Average Latency vs. Number of Nodes

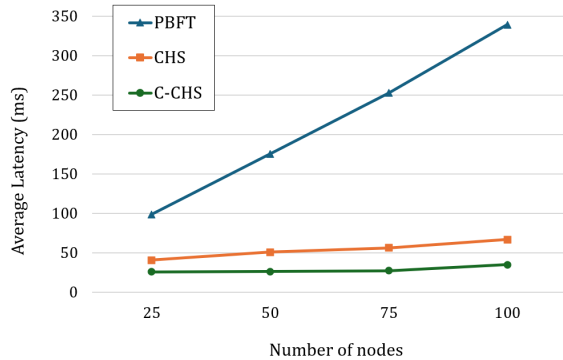


Fig. 5. Average latency comparison across different numbers of nodes.

In terms of latency, Figure 5 shows that C-CHS maintains the lowest values, increasing moderately from 25.78 ms to 34.84 ms. At $n = 100$, C-CHS is significantly faster than both CHS (66.73 ms) and PBFT (339.09 ms). This demonstrates that the combination of committee-based aggregation and leader pre-building systematically shortens the per-round waiting period, leading to substantial latency reductions of up to 90% compared to PBFT.

C. Analysis and Discussion

a) Simultaneous Performance Gains: C-CHS eliminates the leader bottleneck by overlapping the leader's next-block preparation with the committee's QC aggregation. This core mechanism allows C-CHS to simultaneously improve performance compared to CHS, achieving 57-92% higher throughput and 36-52% lower average latency.

b) Superior Scaling and Robustness: Furthermore, C-CHS demonstrates the most stable scaling as the number of nodes increases. In contrast, PBFT's performance degrades sharply due to cumulative broadcast overhead, while CHS is hampered by its leader-centric design. While the absolute performance values are implementation-dependent, the relative improvements and scaling trends were robustly observed, confirming that the proposed design effectively enhances both performance and scalability.

VII. CONCLUSION

This paper proposed a novel IoV-FL framework that organically integrates stack-based deterministic verification, conditional privacy via on-chain pseudonym management, and a

committee-based Chained HotStuff consensus. Through stack scripts, we achieved the standardization of verification rules and lightweight dynamic deployment even in a highly heterogeneous, multi-vendor RSU environment. We also secured a balance between privacy protection and accountability through HID-public key binding and a restricted traceability link. Furthermore, in the consensus layer, we designed the protocol to allow the leader to prepare the next block in parallel while the committee handles vote collection and QC aggregation, which fundamentally reduces leader idle time and enhances processing efficiency.

However, a limitation exists in the potential for additional communication overhead from message exchanges among committee nodes. Accordingly, future work will explore the following: (1) designing a more efficient communication protocol to replace the gossip protocol for QC aggregation within the committee, and (2) strengthening both reliability and efficiency by introducing a reputation-based election mechanism for the leader and committee, instead of the current random selection based on communication stability.

ACKNOWLEDGMENT

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (RS-2023-NR076673). Inshil Doh is the corresponding author.

REFERENCES

- [1] B. Chhetri, S. Gopali, R. Olapojoye, S. Dehbashi and A. S. Namin, "A Survey on Blockchain-Based Federated Learning and Data Privacy," 2023 IEEE 47th Annual Computers, Software, and Applications Conference (COMPSAC), Torino, Italy, 2023, 1311-1318
- [2] Brian Hildebrand, Mohamed Baza, Tara Salman, Simra Tabassum, Bharath Konatham, Fathi Amsaad, Abdul Razaque, "A comprehensive review on blockchains for Internet of Vehicles: Challenges and directions", Computer Science Review, Volume 48, 2023, 100547
- [3] Guanjie Cheng, Junqin Huang, Yewei Wang, Jun Zhao, Linghe Kong, "Conditional Privacy-Preserving Multi-Domain Authentication and Pseudonym Management for 6G-Enabled IoV," in IEEE Transactions on Information Forensics and Security, 2024, vol. 19, 10206-10220
- [4] Ling Xiong, Qiang Li, LeLe Tang, Fagen Li, Xingchun Yang, "Blockchain-based conditional privacy-preserving authentication scheme using PUF for vehicular ad hoc networks", Future Generation Computer Systems, Volume 163, 2025, 107530
- [5] Haibin Zhang, Jiajia Liu, Huanlei Zhao, Peng Wang, Nei Kato, "Blockchain-Based Trust Management for Internet of Vehicles", in IEEE Transactions on Emerging Topics in Computing, vol. 9, no. 3, 2021, 1397-1409
- [6] Maofan Yin, Dahlia Malkhi, Michael K. Reiter, Guy Golan Gueta, and Ittai Abraham, "HotStuff: BFT Consensus with Linearity and Responsiveness", In Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing (PODC '19), 347-356
- [7] Jalalzai, Mohammad M. and Niu, Jianyu and Feng, Chen and Gai, Fangyu, "Fast-HotStuff: A Fast and Robust BFT Protocol for Blockchains", vol. 21, 2024, 2478-2493
- [8] Fei Zhu, Lin You, Jixiang Wang, Lei Li, "P-HotStuff: Parallel BFT algorithm with throughput insensitive to propagation delay", Computer Networks, Volume 262, 2025, 111183
- [9] S. Alqahtani and M. Demirbas, "BigBFT: A Multileader Byzantine Fault Tolerance Protocol for High Throughput," 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC), Austin, TX, USA, 2021, 1-10