

Enhancing Road Safety with Privacy-Preserving Blockchain-Based Mobile Crowdsensing

Zaina Maqour

`zaina_maqour@um5.ac.ma`

Mohammed V University

Hanan El Bakkali

Mohammed V University

Driss Benhaddou

Alfaisal University

Houda Benbrahim

Mohammed V University

Research Article

Keywords: blockchain, mobile crowdsensing, privacy-preserving, intelligent transportation systems, reputation management, road safety, zero-knowledge proofs

Posted Date: October 16th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-11205153/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Enhancing Road Safety with Privacy-Preserving Blockchain-Based Mobile Crowdsensing

Abstract

Road-traffic crashes remain a serious cause of death worldwide, and meeting modern road-safety goals demands dense, trustworthy, real-time data that static and centralised sensing infrastructures cannot provide. Mobile crowdsensing (MCS) is a scalable alternative that captures road telemetry through ubiquitous mobile devices, yet existing systems treat rapid data dissemination and privacy as mutually exclusive. They either sacrifice contributor anonymity to preserve accountability or incur prohibitive on-chain verification costs. This paper presents RBCrowd4RS, a multi-layer, privacy-preserving, blockchain-based MCS system for road safety that resolves these issues. It couples a permissioned consortium ledger with a zk-SNARK anonymous-authentication scheme, in which a citizen proves authorized membership without revealing their real identity. A four-factor Gompertz reputation model integrating on-device AI consistency and automated inactivity decay, coupled with region-gated validator selection, anchors trust and accountability on-chain. Benchmarked via Hyperledger Caliper on an end-to-end prototype, constant 520 byte authentication proofs verify on-chain in 2.75ms across one million users, with local filtering adding just 70.5 ± 11.9 ms. Additionally, reputation weighting caps 100 Sybils at 31% voting power and elevates collusion resistance to 59%, proving that unlinkability, accountability, and real-time performance are simultaneously achievable.

Keywords: blockchain, mobile crowdsensing, privacy-preserving, intelligent transportation systems, reputation management, road safety, zero-knowledge proofs

1 Introduction

Road safety remains one of the most pressing public-health challenges of our time. According to the World Health Organisation (WHO), approximately 1.35 million people die in road-traffic crashes every year [1]. The classical *Safe System* (SS) paradigm, originating with Vision Zero in Sweden [2] and globally endorsed by the WHO, rests on the principle that no one should die or be seriously injured on the road network. Its four traditional pillars (safe roads, safe speeds, safe vehicles, and safe users) are being progressively extended by the rapid growth of Intelligent Transportation Systems (ITS), which add cyber-physical dimensions: safe computation, safe data, and safe

communication [3]. Putting these extended pillars into practice requires trustworthy and real-time road-condition data [4].

Mobile Crowdsensing (MCS) has emerged as a decentralised alternative, enabling continuous data acquisition from smartphones and embedded vehicular sensors, harnessing the computational and sensing power already resident in these devices [5], making it a low-cost complement to fixed IoT infrastructure. MCS platforms can identify and geo-reference road anomalies at city scale [6], and efficient routing architectures such as CREAM [7] have shown that crowd-sourced traffic intelligence can be delivered in near-real time. Despite these advances, three fundamental challenges persist in road-safety MCS. First, *worker privacy*: contributors are reluctant to participate when sensing campaigns may expose daily travel patterns. Incentive and privacy-preservation schemes for vehicular crowdsensing have been studied [8, 9], but most rely on pseudonyms that a motivated adversary can trace. Second, *data trustworthiness*: opening a campaign to untrusted crowds invites noise, fabricated reports, and Sybil attacks. Reputation frameworks for participatory sensing exist [10], yet without a tamper-resistant ledger their scores are easily manipulated. Third, *on-device computational overhead*: privacy-enhancing techniques such as zero-knowledge proofs (ZKPs) or homomorphic encryption are typically too expensive for resource-constrained mobile devices [9], which discourages their adoption at scale.

Recent proposals combine blockchain with crowdsensing but generally treat rapid data dissemination and cryptographic privacy as mutually exclusive goals. Systems that prioritise throughput abandon anonymity guarantees; those that pursue cryptographic rigour pay unacceptable verification latency and on-chain storage costs. No existing framework simultaneously achieves identity unlinkability and a tamper-resistant reputation ledger anchored on a permissioned consortium blockchain. This work addresses that exact gap.

We propose **RBCrowd4RS**, a multi-layer, privacy-preserving, blockchain-based MCS system for road-safety enhancement. It integrates (i) a zk-SNARK anonymous-authentication circuit [11] that provides identity unlinkability within the registered anonymity set via succinct non-interactive arguments of knowledge with $O(1)$ on-chain verification, (ii) a tamper-resistant reputation ledger based on our prior reputation-management work [12], (iii) a lightweight edge classifier [13, 14] for on-device quality pre-filtering to discard low-quality uploads before they consume network and storage resources, and (iv) a region-gated, opt-in validation mechanism that combines geographic relevance with contribution history without forcing system-driven committee assignment.

The main contributions of this work are as follows:

- A multi-layer system that combines a permissioned blockchain with mobile crowdsensing, using modular smart contracts and off-chain IPFS storage to acquire, deduplicate, and anchor real-time road-condition telemetry.
- A PLONK zk-SNARK Merkle-membership scheme providing identity unlinkability with $O(1)$ on-chain verification (< 12 ms verification latency) for anonymity sets scaling up to 2^{32} participants, eliminating identity traceback backdoors.
- A customized formulation of the RBCrowd model [12] that resolves asymptotic range-compression for bounded road-safety telemetry scores.

- A full prototype integration spanning a mobile client and Hyperledger Fabric v2.5.4, empirically evaluating transaction latency, MVCC contention, zk-SNARK overhead, and adversarial resilience.

The remainder of this paper is organised as follows. Section 2 surveys related work. Section 3 presents the system model, its entities, and the layered architecture together with the required preliminaries. Section 4 details the workflow of the proposed system. Section 5 reports the end-to-end experimental evaluation, and Section 6 concludes the paper.

2 RELATED WORK

The evolution from the physical Safe System (SS) paradigm toward cyber-physical road-safety architectures has produced a large and fragmented body of work spanning distributed ledgers, mobile crowdsensing, cryptographic privacy, and trust management. Table 1 summarises the surveyed schemes across the capabilities that define the research gap.

The classical SS paradigm, initiated in Sweden [2] and adopted globally by the WHO [1], rests on four physical pillars: safe roads, speeds, vehicles, and users, under the principle that no road user should be killed or seriously injured. The growth of Intelligent Transportation Systems (ITS) [15] has extended this paradigm with cyber-physical dimensions of safe computation, safe data, and safe communication [3]. This is a shift that regional ITS reviews identify as particularly demanding in contexts that lack localized, open-source platforms [4]. Meeting these cyber-physical objectives requires trustworthy real-time data acquisition within the Internet of Vehicles (IoV). Decentralized ledgers have become a primary integrity mechanism. Sharma et al. [16] integrate roadside IoT sensors with deep learning and blockchain smart contracts to trigger emergency responses automatically without a central mediator. Kang et al. [17] apply reputation and contract theory to optimise consensus management in blockchain-enabled IoV, while Javed et al. [18] make trust explicit on-chain through a smart-contract reputation mechanism for federated learning in O-RAN. Decentralization mitigates single-point-of-failure risks, but recording observations on an immutable shared ledger introduces privacy exposure. Federated-learning models for IoV [19] address this with lightweight signatures and reputation-based node selection over pseudonymous identities; however, such designs retain a traceback capability whereby a reputation server can recover a vehicle’s real identity when misbehaviour is detected. Consequently, network resilience improves while unlinkability is not achieved.

Fixed sensing infrastructure is constrained by deployment cost, which motivates Mobile Crowdsensing (MCS) [5] as a scalable alternative. MCS exploits the sensing and computational capacity already present in smartphones and embedded vehicular units. Capponi et al. [20] survey the MCS design space and identify data trustworthiness, participant privacy, and incentive efficiency as the three recurring obstacles to large-scale deployment. In the road-safety domain, El-Wakeel et al. [6] demonstrate that crowd-sourced smartphone data can identify and geo-reference road-surface anomalies with practical accuracy. Efficient data dissemination is a prerequisite for such systems: the CREAM framework [7] balances uploads between multi-hop roadside-unit paths

and cellular links according to node utility. It reports a delivery ratio of 0.9 on a taxi-trajectory dataset from Rome, compared with approximately 0.6 for roadside-unit-only routing. Such routing-oriented studies optimise throughput but do not address the cryptographic protection required in an open, untrusted sensing environment.

A separate line of work couples data collection with cryptographic incentives to protect contributor privacy. Wang et al. [8] propose SPPIM, which employs zero-knowledge proofs and anonymous credentials within smart contracts to validate participants under a reverse-auction model, scaling rewards by bid, node reputation, and quantified data quality without disclosing the driver’s identity. More broadly, zero-knowledge proofs have become a standard building block for privacy in blockchain systems [21]. Also, blockchain-based crowdsensing schemes increasingly embed anonymous authentication and incentive logic on-chain [9]. A persistent obstacle, however, is the computational cost of these primitives on resource-constrained devices, which limits their practicality for high-frequency mobile sensing. Existing designs therefore tend to privilege either verification efficiency or strong anonymity, but rarely both simultaneously.

Because open crowdsensing admits noisy, fabricated, and Sybil contributions, reputation and trust management are central to data reliability. Huang et al. [10] established the case for reputation systems in participatory sensing, and subsequent IoV work has coupled reputation scoring with consensus and node selection [17, 19]. Reputation-aware trust management for vehicular crowdsensing continues to be refined for accountability and attack resistance [22]. In prior work, we introduced RBCrowd [12], a blockchain-based reputation-management framework that aggregates data consistency, validator feedback, task-completion rate, and contribution frequency through a Gompertz function with an automated decay mechanism, yielding bounded scores that penalise inactivity and resist bad-mouthing and on-off attacks. RBCrowd establishes the reputation model but does not integrate anonymous authentication, on-device quality filtering, or a road-safety sensing pipeline. The present work adopts and extends the RBCrowd scoring model as the trust anchor of a complete, privacy-preserving road-safety crowdsensing architecture. It thereby couples tamper-resistant reputation with cryptographic unlinkability; two properties that the surveyed schemes provide only in isolation.

The reviewed literature integrates zero-knowledge proofs [8, 21] with decentralized intelligence [16, 18, 19] and reputation-based trust [10, 12, 17], forming a strong baseline for secure vehicular networks. Nevertheless, rapid data dissemination [7] and cryptographic privacy [8] are consistently treated as mutually exclusive objectives. This trade-off yields architectures that either forgo unlinkability to preserve accountability [22] or incur prohibitive verification latency during consensus. No surveyed framework simultaneously delivers full unlinkability, constant-cost on-chain verification, tamper-resistant reputation, and on-device quality control. The following sections present RBCrowd4RS to close this gap.

Table 1 Comparison of Related Blockchain and Crowdsensing Systems.

Scheme	Primary focus	Blockchain ledger	Full anonymity / unlinkability	No identity backdoor	Reputation management	On-device quality filter	$O(1)$ verification
SRTMS [16]	Blockchain ITS / emergency response	✓	✗	✗	~	✗	
Kang et al. [17]	Blockchain-IoV consensus	✓	✗	✗	✓	✗	
Javed et al. [18]	Blockchain + federated learning IoV	✓	✗	✗	~	✗	
FL-IoV [19]	Federated learning / node selection	✓	~	✗	✓	✗	
CREAM [7]	Social-IoV data dissemination	✗	✗	✗	~	✗	
El-Wakeel et al. [6]	Road-surface crowdsensing	✗	✗	✗	✗	~	
SPPIM [8]	Privacy-preserving incentives	✓	✓	~	~	✗	
Huang et al. [10]	Participatory-sensing reputation	✗	✗	✗	✓	✗	
RBCrowd [12]	Blockchain reputation management	✓	✗	✗	✓	✗	
RBCrowd4RS	Privacy-preserving road-safety MCS	✓	✓	✓	✓	✓	

¹✓: fully supported; ~: partial or with limitations; ✗: not supported.

3 System Model and Architecture

3.1 Core System Entities

The RBCrowd4RS workflow involves four principal actors, illustrated in Fig. 1.

- **Citizens (data contributors).** Registered participants who capture road-safety events using smartphone sensors and vehicle telematics. Contributions are prioritized according to each citizen’s historical reputation and the cryptographic validity of the submission.
- **Validators.** A self-selected subset of citizens who opt in to validate a given submission on the basis of their reputation and proximity to the reported event. Each pending submission is listed as an available validation task to every citizen who is both co-located in the same region as the reported event and above the minimum reputation threshold; any citizen in that eligible set may then choose whether to open and validate it. Validators vote on the utility of a submission, and their reputation-weighted votes form the Validator Feedback (VF) factor. Because validators are drawn from the citizen population, they are not depicted as a separate node in Fig. 1.
- **Blockchain and off-chain storage.** A permissioned consortium ledger records data commitments and the reputation state, while the InterPlanetary File System (IPFS) [23] holds the bulky payloads off-chain; only content identifiers are anchored on-chain. Both are detailed in Section 3.3.
- **Stakeholders (Data Consumers).** Institutional entities, comprising transportation authorities, police, and emergency medical services, that query validated records to drive road-safety decisions, and that fund the sensing campaign.

3.2 Threat Model

RBCrowd4RS assumes an open-participation, semi-trusted environment: any citizen holding a valid Certification Authority (CA) credential may register and contribute, but registration does not imply honest behaviour thereafter, and the consortium’s blockchain nodes and CA are trusted to follow the protocol yet are not assumed

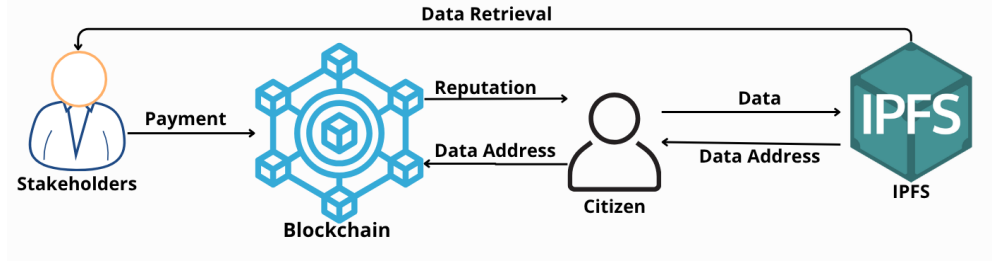


Fig. 1 Interaction model among core system entities in the RBCrowd4RS system.

immune to compromise or coercion. Within this setting, the design specifically targets the following adversary classes.

- **Malicious or dishonest citizens.** A citizen may submit fabricated, low-quality, or irrelevant data to accumulate reputation without genuine effort, or mount an *on-off attack*: behaving honestly over several contributions to build trust, then submitting malicious data once a high reputation score is reached.
- **Malicious or colluding validators.** A validator may submit dishonest feedback to unfairly demote a targeted citizen (*bad-mouthing*) or, together with other validators, collude to accept a deliberately invalid submission. Because RBCrowd4RS does not assume validators are individually trustworthy, feedback is weighted by each validator’s own reputation.
- **Sybil attackers.** An adversary may register many low-cost pseudonymous identities to inflate its aggregate influence over validation outcomes. Every identity starts at the same low initial reputation score, so newly created Sybils carry disproportionately little weight.
- **Honest-but-curious linkage adversaries.** A party that faithfully follows the protocol, whether a stakeholder, a fellow citizen, or an operator of a consortium node, may still attempt to de-anonymise a citizen by correlating on-chain metadata, submission timing, or location patterns across multiple contributions.
- **A compromised or coerced Certification Authority.** Unlike reputation-server designs that retain a real-identity traceback path usable after registration [19], the CA in RBCrowd4RS participates only in the one-time, offline credential-issuance step and is not invoked, consulted, or required to be online during authentication or validation.

3.3 Multi-Layered Architecture Overview

As shown in the architectural diagram in Fig. 2, the system is structured into five horizontal processing layers alongside a vertical, cross-cutting Security Layer.

Sensing Layer. The Sensing Layer forms the physical foundation of the crowd-sensing network. Its primary nodes are registered citizens carrying smartphones and connected vehicles, which opportunistically capture road-safety events such as localized anomalies, potholes, and accidents. Static infrastructure such as roadside cameras and base stations can also feed this layer, but the workflow evaluated in this paper

concerns citizen smartphone image capture; fixed-infrastructure sources are outside the evaluated pipeline.

Edge Layer. Positioned immediately above physical sensing, the Edge Layer acts as an intermediary processing boundary designed to reduce network transmission delays. This layer consists of localized processing units, such as microcontrollers, aggregators, and edge servers. Instead of directly uploading large volumes of raw, noisy telemetry to the IPFS, which would cause severe network congestion, the edge units perform local filtering, data deduplication, and initial anomaly detection.

Data Layer. The Data Layer is responsible for scalable, high-throughput storage of both processed sensing data and analytical results. Due to the computational and financial prohibitiveness of direct on-chain storage, the Data Layer leverages the IPFS as an off-chain distributed storage network. When the edge layer uploads a filtered payload, IPFS returns a unique, content-addressed cryptographic hash known as a Content Identifier (CID). Only this lightweight CID is anchored to the blockchain, which ensures data integrity and traceability while maintaining storage scalability and preventing ledger bloat.

Blockchain Layer. The Blockchain Layer is the immutable, decentralized execution environment of the RBCrowd4RS. Built on a consortium blockchain maintained by pre-authorized entities such as regional transit operators. It manages the system state through three smart contracts: the USERMANAGER (UM) handles anonymous registration and authentication; the DATAMANAGER (DM) validates submissions, enforces a Cuckoo-filter (CKF) deduplication check, and anchors content identifiers; and the REPUTATIONMANAGER (RM) maintains the reputation ledger. Crucially, the layer performs no heavy data processing on-chain; it executes only lightweight cryptographic verification of proofs and CIDs and stores the globally synchronized reputation state.

Decision Layer. At the very top of the architecture, the Decision Layer provides the final interface for data consumers and macro-level decision-makers. The primary stakeholders operating in this layer include municipal transportation departments, police networks, emergency medical services, and academic research institutions. These entities query the verified blockchain records and retrieve the corresponding IPFS payloads to perform macro-level traffic analysis, prioritize road maintenance schedules, and coordinate rapid emergency responses following accidents. Because they derive insights from a transparent, tamper-proof ledger, these institutions can make highly reliable operational decisions without relying on a centralized intermediary.

Security & Privacy Layer. The vertical Security Layer cuts across all horizontal layers to ensure end-to-end trust, data confidentiality, and systemic robustness against adversarial attacks. It comprises three core components:

- **Privacy:** To prevent malicious entities from tracking a citizen’s identity, this module leverages zero-knowledge proofs, specifically Zero-Knowledge Succinct Non-Interactive Arguments of Knowledge (zk-SNARKs). Using zk-SNARKs, data contributors can generate compact, non-interactive proofs to demonstrate the cryptographic validity of their reported data and prove their reputation score without disclosing their identity.

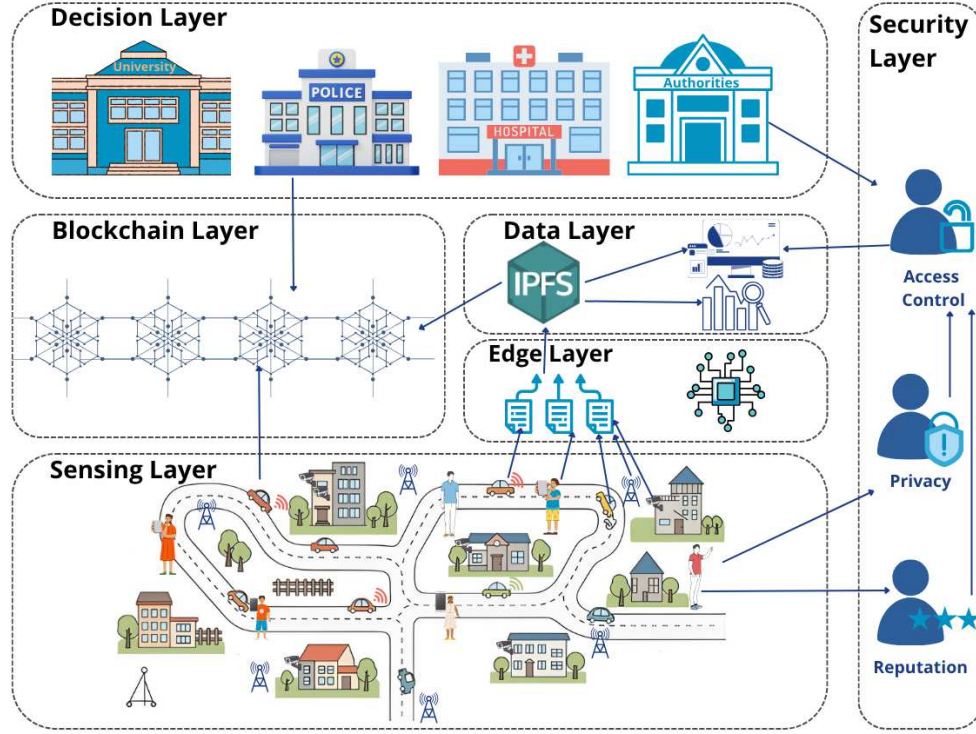


Fig. 2 Layered system architecture illustrating the cross-cutting Security & Privacy Layer.

- **Reputation:** The system integrates our RBCrowd [12] dynamic scoring model to maintain real-time trustworthiness metrics on all citizens. The score aggregates four factors: Data Consistency (DC), Validator Feedback (VF), Task Completion Rate (TCR), and Contribution Frequency (CF). These factors are combined as a weighted sum and passed through a Gompertz growth function with an automated decay mechanism, penalising prolonged inactivity. This dynamic scoring model mitigates malicious behaviour such as bad-mouthing and on-off attacks.
- **Access Control:** A hybrid attribute- and role-based access control (A&R-BAC) model governs which stakeholders may access collected data. Its fine-grained design supports context-aware permissions across distributed edge-cloud environments. It is a separate branch of our research project and is deferred to a companion paper; the present work integrates it only as a background component and focuses on the decentralized reputation ledger and anonymous authentication.

3.4 Preliminaries

3.4.1 Permissioned Blockchain and Smart Contracts

A blockchain is a distributed, immutable ledger maintained by a network of nodes, ensuring data integrity without reliance on a centralized authority. In our context,

public, permissionless blockchains are often unsuitable due to their high consensus latency, transaction fees, and lack of stakeholder privacy. Consequently, the proposed RBCrowd4RS system utilizes Hyperledger Fabric (HLF) [24], an enterprise-grade, permissioned distributed ledger technology (DLT).

Unlike public ledgers, HLF operates on a modular architecture that separates transaction execution from ordering and validation. It allows diverse ITS stakeholders to form a known consortium. HLF employs chaincode, the implementation of smart contracts, to automate system logic upon meeting predefined conditions. Furthermore, HLF’s channel-based architecture provides data isolation, restricting specific road-safety telemetry to authorized nodes and thereby providing the ledger-level substrate on which the deferred A&R-BAC model operates.

3.4.2 PLONK Zero-Knowledge Proof System

The Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK) protocol facilitates the validation of information ownership without disclosing the data itself. Through a non-interactive framework, a prover can demonstrate the validity of their information without the need for direct engagement with the verifier. To overcome the toxic waste vulnerabilities of early zero-knowledge setups, our system leverages the PLONK scheme [11], which utilizes a universal and updatable Structured Reference String (SRS). The mathematical foundation of this protocol is implemented via the following polynomial-time algorithms:

1. $\text{Setup}(1^\lambda) \rightarrow (U, ppk, pvk)$: Using a security parameter λ , a universal SRS U is generated. An arithmetic program C is compiled to produce a public proving key (ppk) and a public verification key (pvk)
2. $\text{GenProof}(ppk, x, w) \rightarrow \pi$: The prover algorithm takes the proving key ppk , a public statement x , and a hidden witness w to generate a zero-knowledge proof π . This proof serves as cryptographic evidence that the prover knows a witness w satisfying the program requirements, while keeping it entirely confidential.
3. $\text{VerProof}(pvk, x, \pi) \rightarrow \{0, 1\}$: The verifier executes this algorithm to assess the validity of π . It returns a Boolean result using the verification key pvk and the public statement x . If the result evaluates to 1, the proof is mathematically sound.

3.4.3 ShuffleNet v2

ShuffleNet v2 [13] is a lightweight convolutional neural network designed for efficient inference on resource-constrained devices. It is guided by direct-metric design principles, balanced channel widths, controlled group convolutions, and reduced network fragmentation which explicitly target real measured latency and memory traffic on the target hardware. The result is a model that, at the x1.0 width, uses roughly 2.3M parameters yet attains classification accuracy competitive with much heavier networks.

We adopt ShuffleNet v2 in the Local Verification step for three reasons aligned with the constraints of mobile crowdsensing. First, its small footprint (under 9 MB in full precision and under 2.5 MB when quantized) fits comfortably on commodity smartphones. Second, its low inference latency on mobile-class CPUs and embedded boards makes the quality check effectively instantaneous relative to the cryptographic

and network operations that follow. Third, by filtering non-road or low-quality images before upload, it conserves both the participant’s battery and the network bandwidth that would otherwise be spent transmitting data that the validators would ultimately reject.

4 Proposed RBCrowd4RS System Workflow

In this section, we describe the proposed scheme in detail, including its operational workflow.

4.1 Anonymous Authentication Scheme

We propose a privacy-preserving authentication scheme where zk-SNARK proof is used to guarantee that the blockchain transactions will not reveal any private information of participants. We specifically implement the PLONK zk-SNARK scheme instantiated over the BN254 elliptic curve, utilizing the gnark cryptographic library. Further, we utilize MiMC as the highly efficient in-circuit hash function and relies on an EdDSA (Baby-Jubjub) signature check to validate the Certification Authority’s credential issuance. This anonymous authentication mechanism is executed through the User Manager (UM) smart contract.

4.1.1 System Setup.

The RBCrowd4RS system is designed to collect a homogeneous type of data, focusing on a single sensing task that is executed repeatedly. Consequently, the same cryptographic parameters can be used at various sensing instances. We adopt a one-time setup phase executed directly by UM contract at deployment. This setup phase includes parameter generation, arithmetic program definition, and key generation. Specifically, the UM generates a universal string U using a security parameter λ and an on-chain randomness beacon that ensures public verifiability and resistance to setup manipulation. The universal SRS U and the compiled circuit C are used to generate a public proving key (ppk), which is distributed to other nodes, and a constant 34.3 kB public verification key (pvk), which is immutably stored in the UM.

4.1.2 Anonymous Registration

The Anonymous Registration phase is a one-time gateway that establishes a citizen’s eligibility while unlinking their real identity from their future sensing submissions. We assume all citizens are welcome to contribute to the sensing task. They need only to apply for an anonymous credential when they want to register on the system. The complete workflow, as detailed in Algorithm 1, is divided into three primary stages: Local Initialization, Offline Certification, and On-Chain Registration.

First, the registration process starts by locally selecting a high-entropy random secret $s \in \mathbb{Z}_q$. This value represents the witness for all future zero-knowledge proofs and is never shared with any entity. Using the in-circuit MiMC algorithm, the device computes a cryptographic commitment $H = \text{Hash}_{MiMC}(s)$. This commitment H represents the user’s anonymous anchor on the blockchain.

During the offline Certification Phase, the citizen provides their real-world identity (e.g., National ID) and commitment H to the Certification Authority (CA). Upon successful verification, the CA issues a digital signature $\sigma_{CA} = \text{Sign}_{EdDSA}(H)$, which represents the certificate of eligibility.

The citizen then utilizes the ppk and their private witness $w = \{s, \sigma_{CA}\}$ to generate a succinct zk-SNARK proof, π_{reg} . This circuit compiles to exactly 13,799 PLONK constraints. Generating this proof takes an average of 661 ms on the client side, producing a compact proof size of exactly 520 Byte. Finally, the UM contract receives the proof π_{reg} and executes the $\text{VerProof}(pvk, H, \pi_{reg})$ function. If the proof is valid, the contract inserts H into a Merkle Tree Ω stored on the ledger.

Algorithm 1: Anonymous Registration

Input: S, ppk, pvk
Output: updated Ω

```

/* Local Initialisation */
1 Citizen selects high-entropy secret  $S \in \mathbb{Z}_q$  locally;
2  $H = \text{Hash}_{MiMC}(s)$ ;

/* Certification Phase (Offline) */
3 Citizen provides  $\{\text{RealIdentity}, H\}$  to  $CA$ ;
4  $CA$  verifies identity;
5  $\sigma_{CA} = \text{Sign}_{EdDSA}(H)$ ;

/* On-Chain Registration */
6  $\pi_{reg} = \text{GenProof}(ppk, H, \{s, \sigma_{CA}\})$ ;
7  $UM$  receives  $\{\pi_{reg}, H\}$  and executes verification;
8 if  $\text{VerProof}(pvk, H, \pi_{reg}) == 1$  then
9   | Insert  $H$  into  $\Omega$  ;
10  | return "Registration Successful";
11 else
12  | return "Failure";

```

4.1.3 Session-Based Anonymous Authentication

This phase is where citizens actually start contributing data through the mobile application. When a mobile device detects a road anomaly, it must authenticate the session without exposing the user's real identity. Revealing the registration commitment H on every session, as a naive scheme might, would let the Certification Authority, which learns H at issuance, attribute every session to a real person. RBCrowd4RS instead publishes a context-bound pseudonym that is computationally independent of H , and proves its validity entirely inside the zk-SNARK circuit. The complete authentication workflow is detailed in Algorithm 2.

On the client side, the device first generates a temporary, one-time session key pair (sk_{sess}, pk_{sess}) . It then derives its pseudonym $P = \text{Hash}_{MiMC}(s \parallel ctx)$, binding

the citizen's private secret s to ctx , a public context value published by the UM contract and held constant for a deployment epoch, using the same MiMC hash as the registration commitment H . Because ctx is fixed by the contract rather than chosen by the client, each citizen holds exactly one pseudonym per epoch: P is the stable identifier under which the RM accumulates the trust factors, and a single credential cannot be turned into several on-chain identities. At the same time, P cannot be linked to H , and hence to the citizen's real identity, by any party that does not know s , the certification authority included. Rotating ctx at an epoch boundary would additionally sever linkage between epochs, at the cost of carrying reputation across the rotation; we leave anonymous reputation carry-forward to future work. The device then constructs the authentication proof

$$\pi_{auth} \leftarrow \text{GenProof}(ppk, \underbrace{\{root, P, ctx, pk_{sess}\}}_{\text{public}}, \underbrace{\{s, \sigma_{CA}, \pi_{\Omega}\}}_{\text{witness}}),$$

where $root$ is the Merkle root the citizen believes is currently stored on-chain and π_{Ω} is the citizen's own Merkle inclusion path for $H = \text{Hash}_{\text{MiMC}}(s)$. The circuit constraints enforce, without revealing s , σ_{CA} , or π_{Ω} , that (i) $H = \text{Hash}_{\text{MiMC}}(s)$ is a valid leaf of the Merkle tree rooted at $root$, reached via π_{Ω} ; (ii) H carries a valid CA signature σ_{CA} established at registration; and (iii) the declared pseudonym satisfies $P = \text{Hash}_{\text{MiMC}}(s \parallel ctx)$. Condition (i) is the depth-20 Merkle inclusion proof already accounted for in the authentication circuit's constraint count. The payload $\{\pi_{auth}, P, pk_{sess}\}$ is then uploaded to the UM smart contract; neither H nor s ever leaves the device.

On-chain, the UM contract fetches its own current Merkle root rather than trusting a client-supplied value, and first checks P against the set $\mathcal{N}$ of pseudonyms that currently hold an open session. If $P \in \mathcal{N}$ the request is rejected, so a citizen holds at most one live session at a time; entries are removed from $\mathcal{N}$ when a session expires or is closed. If $P \notin \mathcal{N}$, the contract executes $\text{VerProof}(pvk, \{root, P, ctx, pk_{sess}\}, \pi_{auth})$. Replay protection does not depend on $\mathcal{N}$: because pk_{sess} is a public input, a captured proof verifies only for the session key it was generated with, and every subsequent submission must be signed with the matching sk_{sess} , which an eavesdropper does not hold. Despite the added Merkle-path and pseudonym constraints, the succinctness of the PLONK scheme keeps the proof at exactly 520 Byte, and on-chain verification completes in 2.75 ms. If the proof is valid, the UM registers pk_{sess} under P , inserts P into $\mathcal{N}$ for the lifetime of the session, and grants the citizen authorization to upload their data to IPFS; if the proof fails, the session is denied.

4.2 Data Submission and Verification

After successful authentication, a citizen obtains an active session to submit their data. This phase ensures the integrity and quality of sensing contributions. Before initiating any checks, the existence of an active session is verified. If no active session is found, the submission is locally rejected. Once a session check is validated, data submission and verification processes are initiated. The process of this phase involves three steps: local verification, data submission, and validator evaluation. These steps

Algorithm 2: Anonymous Authentication

Input: s, σ_{CA}, ppk, pvk , Merkle path π_Ω
Output: Session Status $\{1, 0\}$

```
/* Client side */
1 Device generates a temporary session key pair  $(sk_{sess}, pk_{sess})$ ;
2  $ctx \leftarrow$  current epoch context published by  $UM$ ;
3  $P \leftarrow \text{Hash}_{MiMC}(s \parallel ctx)$ ;
4  $\pi_{auth} \leftarrow \text{GenProof}(ppk, \underbrace{\{root, P, ctx, pk_{sess}\}}_{\text{public}}, \underbrace{\{s, \sigma_{CA}, \pi_\Omega\}}_{\text{witness}})$ ;
5 Upload  $\{\pi_{auth}, P, pk_{sess}\}$  to  $UM$ ;
/* On-chain verification by smart contract */
6  $root \leftarrow$  current Merkle root stored in  $UM$ ;
7 if  $P \in \mathcal{N}$  then
8   return 0 ; // pseudonym already has an open session
9 if  $\text{VerProof}(pvk, \{root, P, ctx, pk_{sess}\}, \pi_{auth}) = 1$  then
10   Register  $pk_{sess}$  under  $P$  in  $UM$ ; insert  $P$  into  $\mathcal{N}$  until session expiry;
11   return 1 ; // session opened successfully
12 return 0 ; // proof verification failed
```

are implemented using different functions in the Data Manager (DM) smart contract. Algorithm 3 details these steps.

4.2.1 Local Verification

Once authentication is ensured, the citizen is authorized to capture the detected road anomaly. To preserve network bandwidth and minimize on-chain processing costs, RBCrowd4RS performs pre-upload verification to filter inconsistent road data. The data quality is determined as inspired from our previous work [12], specifically the Data Consistency (DC) factor. This factor considers data quality as proportional to its similarity to other contributions in the same sensing task. Given that our system considers collecting road data as the only sensing task, this factor ensures that the uploaded image actually contains road-related information (e.g., potholes, cracks, or signs).

To this end, an AI model is implemented to run locally when a citizen captures an image of a road anomaly. Once the data quality report is generated: if the output is less than 0.5, the sensed data is rejected; otherwise, the data is uploaded.

4.2.2 Decentralized Storage and Deduplication

The sensed data is never stored directly on the blockchain. Instead, only a cryptographic reference and a fingerprint are handled by the smart contract. The encrypted data is uploaded to IPFS, which returns a Content Identifier (CID), a unique cryptographic hash of the data.

To prevent the same data from being processed and rewarded multiple times, the DM smart contract maintains a Cuckoo Filter (CKF) on-chain. The DM contract generates a fingerprint of the incoming report and checks the CKF; if found, the data is likely redundant. In this case, the submission is rejected to avoid redundant endorsement, ordering and storage cost. If not found, the report is unique. The fingerprint is inserted into the filter, and the process continues.

4.2.3 Validator Evaluation

After the data consistency check, this phase ensures that submitted data are verified by reputation-weighted validation. Inspired by the dynamic trust management in RBCrowd [12], RBCrowd4RS does not have the system pick who validates; instead, the DM makes the submission discoverable to citizens qualified to judge it and lets them decide whether to participate. Candidates are restricted to citizens who meet a minimum reputation threshold and are co-located in the same region as the reported event; the DM lists the submission to every citizen satisfying both conditions, and each independently chooses whether to open the task, retrieve the encrypted data from IPFS, and submit feedback. The resulting validator set V is therefore the variable-size subset of eligible citizens who opted in, not a fixed-size committee assigned by the system. Each opted-in validator assigns a binary feedback score $\{-1, +1\}$, reflecting their satisfaction with the quality and utility of the submitted data, weighted by the validator’s reputation R_v so that reliable validators have greater influence.

4.3 Dynamic Reputation Management

Once the data is accepted, the Reputation Manager (RM) smart contract is triggered to update the citizen’s reputation score. Reputation is stored under the citizen’s pseudonym P , which is stable across sessions within an epoch; the RM therefore accumulates each citizen’s history without learning, or needing, the citizen’s real identity. The reputation scheme used in RBCrowd4RS is a direct application of our previously published RBCrowd framework [12] to the single-task, road-anomaly-sensing context of this paper. RBCrowd defines five trust factors, Validator Feedback (VF), Data Consistency (DC), Contextual Alignment (CA), Task Completion Rate (TCR), and Contribution Frequency (CF), aggregated through a Gompertz function. Because RBCrowd4RS collects one homogeneous task type (road-anomaly images) rather than RBCrowd’s heterogeneous task pool, the Contextual Alignment factor, which checks submissions against task-specific metadata, does not apply; RBCrowd4RS reuses RBCrowd’s four remaining factors, with their exact formulations recalled below.

4.3.1 Validator Feedback (VF) Factor

As introduced in Section 4.2, each opted-in validator $v \in V$ assigns a binary feedback score $f_{v \rightarrow i} \in \{-1, +1\}$ to citizen i ’s submission. Following RBCrowd [12], the raw validator-feedback score weights each vote by the validator’s own reputation R_v , and

Algorithm 3: Data Submission and Validation

Input: $D, L, sk_{sess}, pk_{sess}, \mathcal{CKF}$
Output: Contribution Status $\{1, 0\}$

```
/* Step 1: Local Verification */
1 if Road Anomaly Detected then
2   if  $pk_{sess}$  is NOT active OR Session Expired then
3     return 0; // Session Inactive - Please Re-authenticate
4   else
5      $DC \leftarrow \text{ShuffleNetV2}(D)$ ;
6     if  $DC < 0.5$  then
7       return 0; // Inconsistent Data Reject
8      $CID \leftarrow \text{IPFS\_Upload}(\text{Encrypt}(D))$ ;
9      $\sigma_{sess} \leftarrow \text{Sign}_{sk_{sess}}(CID \parallel L)$ ;
10    Send  $\{CID, L, pk_{sess}, \sigma_{sess}\}$  to DM;

/* Step 2: On-Chain Verification */
11 if  $pk_{sess}$  is not registered as an open session in UM then
12   return 0; // Unknown or expired session
13 if  $\text{Verify}_{pk_{sess}}(\sigma_{sess}, CID \parallel L) \neq 1$  then
14   return 0; // Invalid session signature
15  $f \leftarrow \text{Hash}(CID, L)$ ;
16 if  $\text{Query}(\mathcal{CKF}, f) == \text{True}$  then
17   return 0; // Duplicate Report Detected - Rejected
18 Insert( $\mathcal{CKF}, f$ );

/* Step 3: Validator Evaluation */
19  $E \leftarrow \text{EligibleCitizens}(\text{Reputation} \geq \theta_{rep}, \text{Region}(L))$ ;
20 List submission to  $E$ ;
21  $V \leftarrow \{v \in E : v \text{ opts in}\}$ ;
22 foreach  $v \in V$  do
23    $f_{v \rightarrow i} \leftarrow \text{BinaryFeedback}(v, CID)$ ;
24  $VF_i \leftarrow \sum_{v \in V} R_v \times f_{v \rightarrow i}$ ;
25 if  $|V| \geq q_{min}$  and  $VF_i > 0$  then
26   return 1; // Accepted
27 return 0; // Rejected or quorum not reached
```

thresholded at zero to remove the effect of negative cumulative feedback:

$$VF_i = \sum_{v \in V} R_v \cdot f_{v \rightarrow i}, \quad (1)$$

4.3.2 Data Consistency (DC)

In the original RBCrowd framework [12], the Data Consistency (DC) factor was formulated for heterogeneous, multi-task crowdsensing. It evaluated how closely a worker’s submission d_i agreed with peer contributions on the same task via pairwise similarity metrics. RBCrowd4RS does not employ this pairwise equation directly because it focuses on a single sensing task: road-surface anomaly detection. Instead, we adapt the DC factor into an objective, model-driven metric that assesses data consistency directly against benchmark road anomaly distributions. Specifically, consistency is measured locally at the edge using the lightweight ShuffleNet v2 classifier [13]. If the model’s output confidence for the captured image d_i meets or exceeds the system acceptance threshold ($DC_i \geq 0.5$), the sensed data is accepted for upload, and the continuous score DC_i is committed on-chain as the data consistency component of citizen i ’s reputation update.

4.3.3 Task Completion Rate (TCR) Factor

TCR reflects the historical reliability of citizen i across their set of submissions S_i , weighted by each task’s quality and the citizen’s prior reputation R_i^{old} .

$$TCR_i = \frac{1}{|S_i|} \sum_{k \in S_i} DC_i^k \sqrt{R_i^{old}}, \quad (2)$$

normalised as $\max\{0, TCR_i\}$. The $\sqrt{R_i^{old}}$ term introduces a diminishing-return effect that rewards consistent engagement while preventing runaway reputation inflation [12].

4.3.4 Contribution Frequency (CF) Factor

CF rewards regular participation and penalises long gaps. Let $t_1, \dots, t_n$ denote the rounds at which citizen i has contributed; the mean inter-arrival time is

$$\overline{\Delta t_i} = \frac{1}{n-1} \sum_{k=2}^n (t_k - t_{k-1}), \quad (3)$$

and the contribution-frequency factor is

$$CF_i = e^{-\beta \overline{\Delta t_i} / \Delta t_{max}}, \quad (4)$$

reused exactly from RBCrowd [12], where Δt_{max} is the maximum tolerable inactivity window and β controls the sensitivity of decay to irregular participation.

4.3.5 Gompertz Reputation Aggregation

The four factors are combined into a raw weighted score,

$$\hat{R}_i = \gamma_1 VF_i + \gamma_2 DC_i + \gamma_4 TCR_i + \gamma_5 CF_i, \quad (5)$$

With $\sum \gamma_k = 1$. To prevent uncontrolled reputation growth and ensure diminishing returns for repeated contributions, the aggregated score is passed through a Gompertz function [10],

$$R_i = e^{-e^{-\lambda \hat{R}_i}} \quad (6)$$

Every parameter, λ , β , Δt_{max} , and the relative weight of each factor, is reused directly from RBCrowd’s published values.

5 Performance Evaluation and Discussion

We evaluate RBCrowd4RS along the three axes that determine its practical viability: (i) the cost of the zero-knowledge proof system that underpins anonymous authentication, (ii) the throughput and latency of the permissioned ledger that anchors reputation and submission state, and (iii) the security and fairness properties of the reputation-weighted incentive mechanism.

5.1 Experimental Testbed and Methodology

The prototype is deployed on a commodity x86-64 workstation (Intel Core i7-12700K, 16GB RAM, Windows 11). All services run as Docker containers under Docker Desktop v4.73.1 with the WSL2 backend. The ledger tier runs Hyperledger Fabric 2.5.4 [24] configured as a single organization with one endorsing peer backed by CouchDB and one Raft ordering node; the three application chaincodes (USERMANAGER, DATA-MANAGER, REPUTATIONMANAGER) are deployed as external chaincode services in Go.

The citizen-facing mobile client is a React Native application (RN 0.81.5, React 19.1.0) built on Expo SDK $\sim 54.0.0$ and run through the EXPO GO companion app. The client was exercised on an Infinix X6531 (Infinix Hot 50i) smartphone, a MediaTek Helio G81 device running Android 14 (XOS 14.5) with 6GB of memory, and captured images are compressed to 1024×768 px on-device before upload to conserve radio bandwidth.

The zero-knowledge tier is implemented with the **gnark** library [25] using the PLONK proving scheme [11] over the BN254 elliptic curve with a universal (updatable) KZG structured reference string, and MiMC [26] as the in-circuit hash. Each reported proving and verification figure is the mean over 20 timed iterations after a warm-up run. The ledger load tests are executed with the Hyperledger Caliper benchmark suite [27], using a closed-loop generator that maintains a fixed number of in-flight transactions until a target operation count completes.

The reputation properties are evaluated with a faithful re-implementation of the on-chain scoring logic driven by Monte-Carlo simulation with a fixed random seed for reproducibility. The participation experiments place a static population of 300 citizens i.i.d. uniformly at random over a $0.5^\circ \times 0.5^\circ$ bounding box ($\approx 55 \times 55$ km) centered on a metropolitan reference point (Casablanca, Morocco: 33.5899°N , -7.6039°W); citizen locations do not move during a run, so no mobility model is applied. Each of the 3,000 simulated submission events is drawn i.i.d. uniformly within the same bounding box. For every event, a citizen is eligible to be listed as a candidate validator

Table 2 Performance and Circuit Complexity of zk-SNARK Primitives

Circuit	Constr. (gates)	Setup (ms)	Prove (ms)	Verify (ms)	Proof (B)	PK (MB)
Registration	13,799	870	661±347	5.42	520	1.08
Authentication	18,223	2239	1396±314	2.75	520	2.13

if their reputation is at or above $\theta = 0.5$ and their haversine great-circle distance to the event is at or below a fixed region radius of 6 km; each eligible citizen then opts in independently with probability $p = 0.45$, a disclosed modelling assumption standing in for the real-world opt-in rate, which is not yet observable from a deployment. This synthetic domain is a simplification of real pedestrian and vehicular mobility and of true opt-in behaviour, adopted to isolate the locality effect of region-gating itself; calibrating p against a real deployment and evaluating under a realistic mobility trace are left as future work.

5.2 Cryptographic Benchmarks of the zk-SNARK Engine

Table 2 reports the cost of the two production circuits. The *Registration* circuit enforces a MiMC commitment opening together with an EdDSA (Baby-Jubjub) signature check binding the commitment to the Certification Authority, and compiles to 13,799 PLONK constraints. The *Authentication* circuit enforces the same commitment opening plus a depth-20 Merkle inclusion proof, and compiles to 18,223 constraints. Despite this difference, the two circuits produce proofs of identical size (520 byte) and share a constant verifying key (34.3 kB), a direct consequence of the succinctness of the scheme: the on-chain verifier cost is independent of the statement complexity.

Proof generation, which executes on the citizen’s device, is the dominant client-side cost at 0.66 s for registration and 1.40 s for authentication. Crucially, verification, which executes inside the smart contract on every authentication, completes in 2.75 ms for the production authentication circuit and never exceeds 12 ms across the entire range we tested. The one-time universal setup (0.87 s and 2.24 s respectively) is amortized over the lifetime of the deployment and is performed once at contract initialization.

To characterize how the authentication proof scales with the size of the anonymity set, Fig. 3 sweeps the Merkle tree depth from 4 to 32, corresponding to anonymity sets of 2^4 to 2^{32} registered citizens. Constraint count and proving time grow linearly with depth (from 3,999 constraints / 0.25 s at depth 4 to 28,891 constraints / 1.02 s at depth 32). In contrast, Fig. 4 shows that verification time and proof size remain essentially flat below 12 ms across all depths. The production depth of 20 therefore supports more than one million anonymous participants while keeping on-chain verification in the single-digit-millisecond range, confirming that the scheme imposes no asymptotic on-chain penalty as the user base grows. The relatively high standard deviation on proving time (Table 2) is an artifact of co-locating the prover with the full Docker-based ledger stack on one machine; the reported means remain stable across runs.

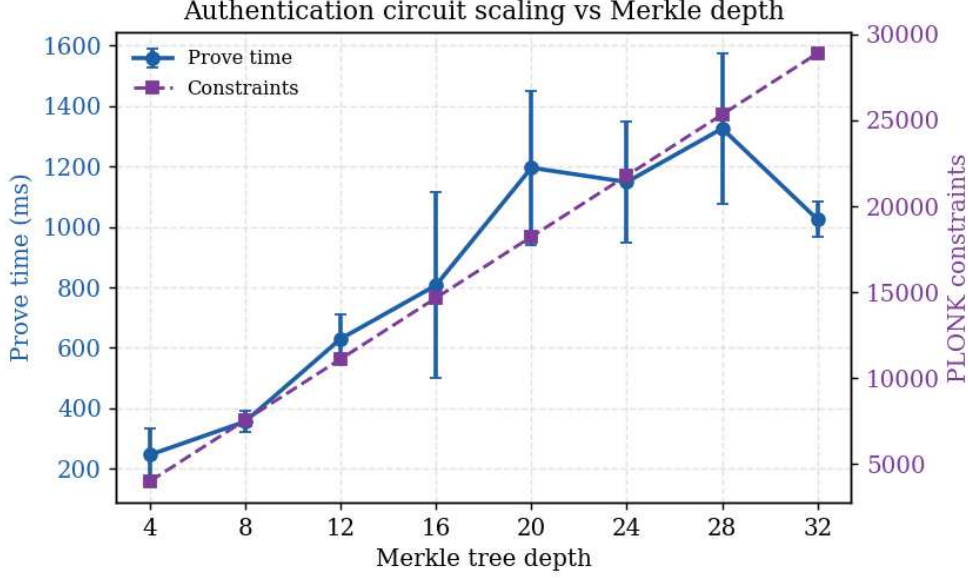


Fig. 3 Proving latency and PLONK constraint scaling across Merkle tree depths.

Table 3 Ledger Query Throughput and Latency Percentiles (GetReputation)

Clients	TPS	Mean (ms)	p50 (ms)	p95 (ms)	p99 (ms)
1	33.2	30.1	15.0	54.0	54.0
2	115.5	17.3	12.0	56.3	56.3
4	180.1	22.1	16.0	37.2	60.0
8	236.6	33.6	21.0	59.7	74.0
16	214.8	72.8	49.0	121.7	148.0
32	227.4	135.8	103.0	216.5	250.0
64	257.7	232.6	161.0	357.4	406.0

5.3 Ledger Throughput, Scalability, and Latency

We separate the ledger workload into a query (read) path and an invoke (write) path, since they exercise different parts of the Fabric pipeline. The read path (`GetReputation`) is satisfied entirely by peer endorsement and does not involve ordering, whereas the write path (`InitReputation`) traverses the full endorse-order-commit cycle.

Fig. 5 and Table 3 report the read path under increasing client concurrency. Throughput rises steeply up to eight concurrent clients (236 tx/s) and reaches a peak of 258 tx/s at 64 clients, while the median latency stays below 70 ms until 16 clients before growing linearly as the system saturates. No read operation failed at any concurrency level. This confirms that reputation lookups, which occur on every validator-eligibility check and reward computation, are not a bottleneck.

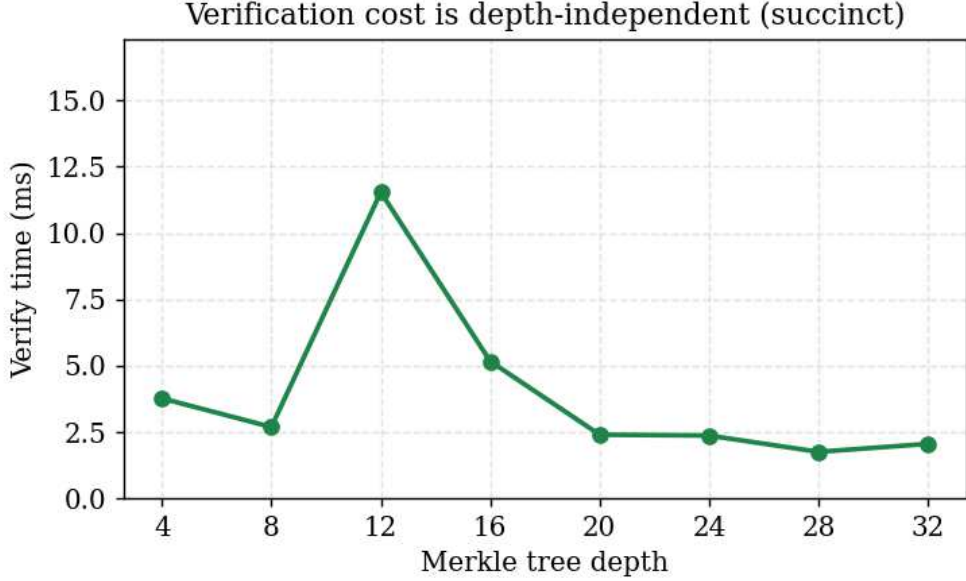


Fig. 4 On-chain zk-SNARK verification latency across varying Merkle tree depths.

The write path, shown in Fig. 6, exposes two distinct limiting factors that are instructive for any reputation-bearing ledger design. First, at a single client the end-to-end commit latency is 2.1 s, which matches the orderer `BatchTimeout` of 2 s: an isolated transaction must wait for the batch timer to expire before its block is cut. This floor is a tunable configuration parameter, not a fundamental limit, and can be lowered for latency-sensitive deployments. Second, as concurrency increases, the `InitReputation` transactions begin to conflict on a single shared state key, the global leaderboard index that orders all registered commitments. The resulting `MVCC_READ_CONFLICT` rate climbs from 0% at one client to over 90% at 32 clients, capping effective write throughput at roughly 4.5 tx/s. This is a textbook hot-key contention pattern; it identifies the global index as the single point of write serialization and motivates a clear optimization for production, namely sharding the leaderboard index across composite-key ranges or updating it asynchronously off the critical path. We report this result transparently because it reflects an application-level design choice rather than a limitation of the underlying consensus.

5.4 Edge AI Inference Performance

5.4.1 Implementation

The Local Verification step is implemented as a ShuffleNet v2 x1.0 backbone [13] with a custom classification head, $1024 \rightarrow 256 \rightarrow 2$ with ReLU and dropout ($p = 0.3$), replacing the network’s original 1000-class ImageNet output layer with a binary road/non-road decision. The PyTorch model is exported through an ONNX intermediate representation to a float16-quantised TFLite flatbuffer for on-device execution, targeting the

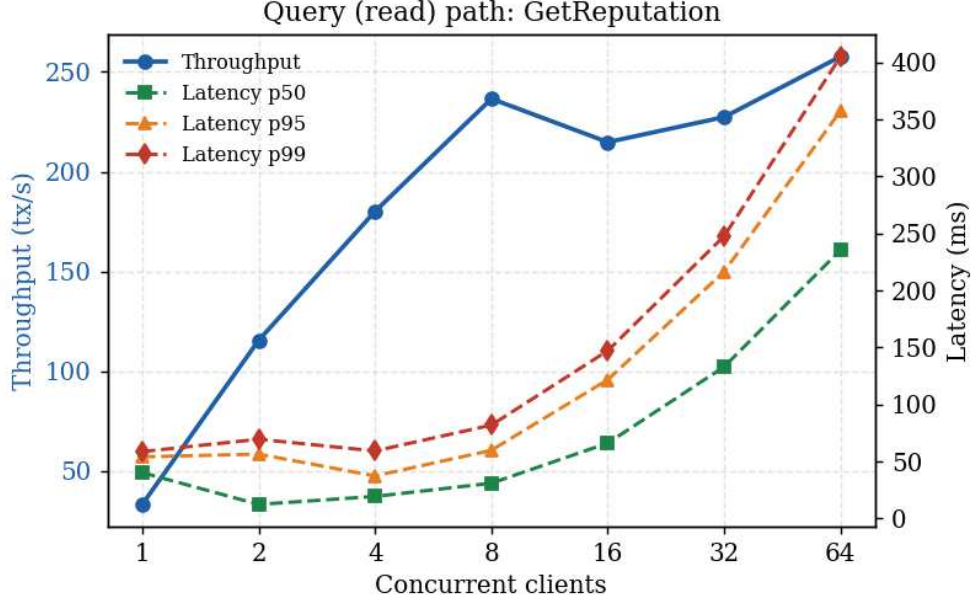


Fig. 5 Read-path throughput and response latency percentiles under varying client concurrency (GetReputation).

mobile client. At the time of writing, the backbone is initialised from ImageNet-pretrained weights and the export pipeline is implemented and verified end-to-end; the classification head has not yet been fine-tuned on labelled road-anomaly imagery, so this subsection reports inference-cost measurements only and makes no classification-accuracy claim. Fine-tuning the head on a labelled road-anomaly dataset and bundling the exported TFLite model with the mobile client are ongoing work.

With the custom classification head, the resulting architecture contains 1.52 M parameters, compared with approximately ≈ 2.3 M parameters for the original 1000-class ShuffleNetV2 model [13]. The reduction is primarily due to replacing the original ImageNet classifier with the substantially smaller two-class head.

5.4.2 Benchmark Latency and Cross-Platform Comparison

Because forward-pass latency is determined by the network’s architecture (layer shapes, channel counts, operation count) rather than by specific weight values, it can be measured honestly on the current checkpoint even though the head is not yet fine-tuned. We benchmark the model following the same warm-up-then-time methodology used for the zk-SNARK benchmarks: 10 warm-up passes followed by 50 timed single-image ($224 \times 224 \times 3$) forward passes. Fig. 7 shows the resulting single-thread latency distribution: mean 70.5 ± 11.9 ms, median 71.8 ms, and 95th percentile 90.5 ms. Pinning inference to all available threads is, counter-intuitively, *slower* on this model (mean 104.3 ms): at 1.52 M parameters the tensors are small enough that multi-threading

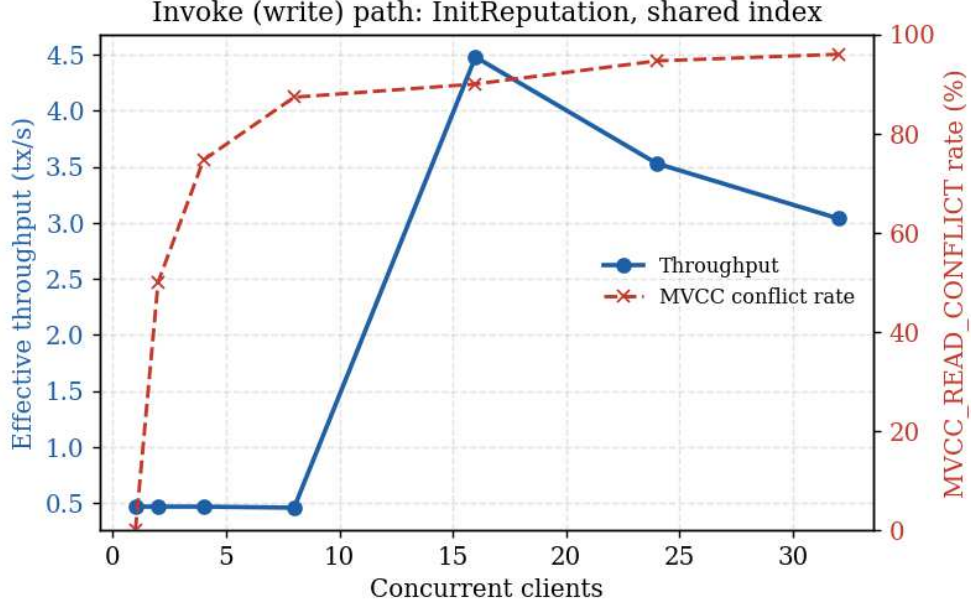


Fig. 6 Write-path throughput and MVCC read conflict rate under concurrent client load (InitReputation).

overhead outweighs the benefit of parallel execution, a known effect for lightweight networks.

Fig. 8 places this measurement alongside representative mobile- and embedded-class figures already reported in the literature. On an embedded GPU platform (NVIDIA Jetson Nano), ShuffleNet v2 infers a single sample in 16.96 ms [28]; on a CPU-only single-board computer (Raspberry Pi 4B), reported latencies range from roughly 105 ms with optimised runtimes to 280 ms under an unoptimised PyTorch stack [14]. Our measured 70.5 ms single-thread figure sits between these two literature references, outperforming a Raspberry Pi’s ARM core while remaining well short of a dedicated embedded GPU.

Two architectural observations follow. First, the local filtering delay, 16-105 ms across the measured and optimized platforms, is at most a fifth of, and typically under a twentieth of, the 1.40 s spent generating the authentication proof; it is therefore a negligible addition to the end-to-end submission path once deployed. Second, running this lightweight model locally remains far cheaper than uploading raw data: a single high-resolution road image costs several megabytes of radio transmission, which on a smartphone dominates the energy of a sub-second on-device CNN inference. Discarding false anomalies at the edge avoids both this radio energy and the downstream IPFS, ordering, and validation work that an invalid submission would otherwise trigger.

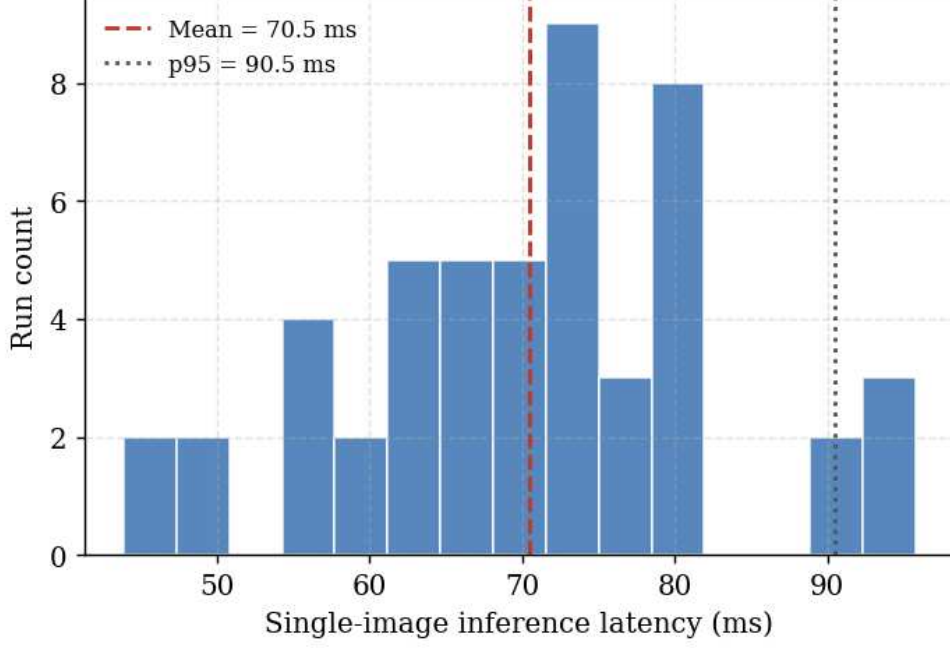


Fig. 7 Empirical distribution of single-thread inference latency for the on-device ShuffleNet v2 model.

5.5 Region-Gated Validator Selection

Unlike a system that samples a committee for a citizen, RBCrowd4RS lists each pending submission to every citizen who is simultaneously within the fixed region radius of the event and at or above the reputation eligibility threshold, and lets each of them independently decide whether to open and validate it. We simulate 3,000 such submissions against the 300-citizen population described in Section 5.1 and evaluate two questions: how many validators a typical submission attracts, and how much the region gate improves locality relative to drawing the same number of validators from the whole, ungated citizen pool.

Fig. 9(a) shows the distribution of opted-in committee sizes: the mean is 4.29 and the median is 4, and 79.0% of submissions reach the minimum quorum of 3 validators we require before a verdict is accepted; the remainder need either a wider region, a higher opt-in rate, or a longer collection window before finalisation, all deployment-tunable parameters. Fig. 9(b) confirms the locality benefit of gating: validators who actually opt in sit at a median of 4.2 km from the event, against 24.9 km for an equal-size draw from the ungated pool, a $5.97\times$ reduction, because the region radius mechanically excludes distant citizens rather than merely down-weighting them. This locality is a structural property of the gate itself, not an artifact of a selection algorithm, and holds regardless of how many eligible citizens ultimately choose to respond.

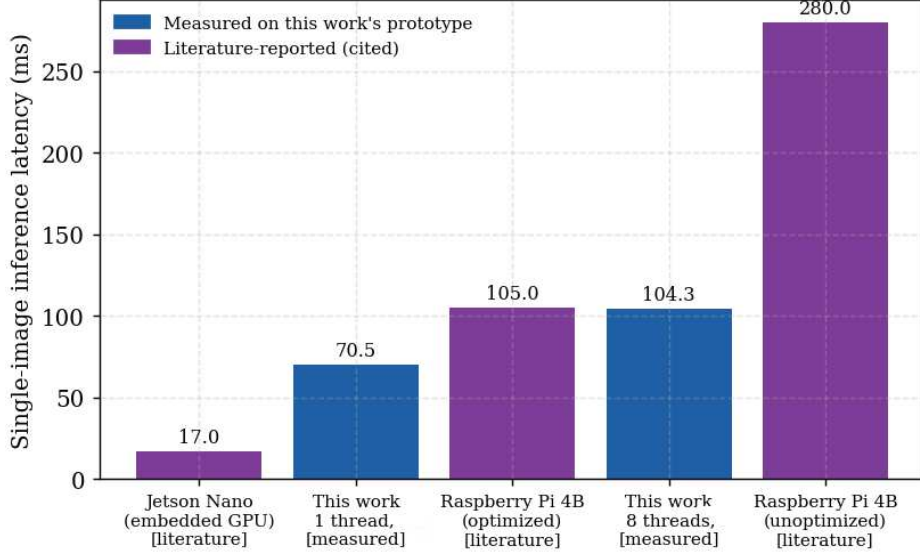


Fig. 8 Single-image inference latency across measured and literature-reported platforms.

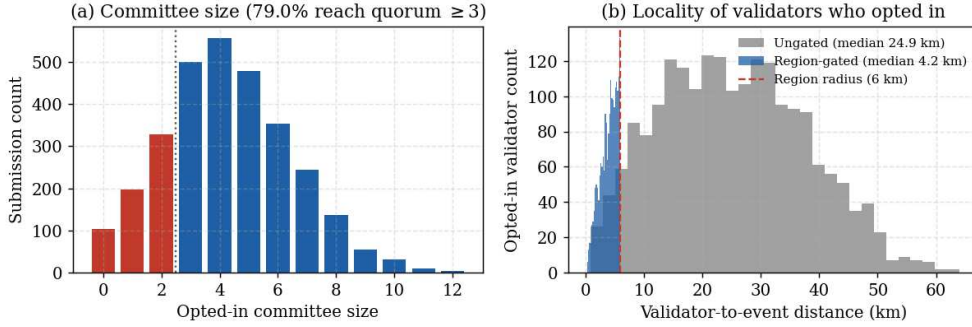


Fig. 9 Evaluation of region-gated validator selection: (a) distribution of opted-in committee sizes; (b) validator-to-event spatial locality comparison.

5.6 Reputation Dynamics and Adversarial Resilience

We next evaluate the reputation engine, whose four factors (Data Consistency, Validator Feedback, Task Completion Rate, and Contribution Frequency) are aggregated per Eq. (5) and passed through the Gompertz mapping of Eq. (6) to produce the final score $R_i \in (0, 1)$. Every newly registered citizen is assigned RBCrowd’s explicit initial score $R_0 = 0.5$. Fig. 10 plots this mapping: neutral behaviour ($\hat{R}_i = c = 0.5$) reproduces the landmark $G(c) = 1/e \approx 0.368$, while the score approaches 0 and 0.921 at the domain extremes, enforcing the intended diminishing returns so that no participant can grow unbounded influence through sheer volume.

Because reputation also depends on sustained activity, we further verify the inactivity-decay mechanism: once a worker stops contributing, its last score decays

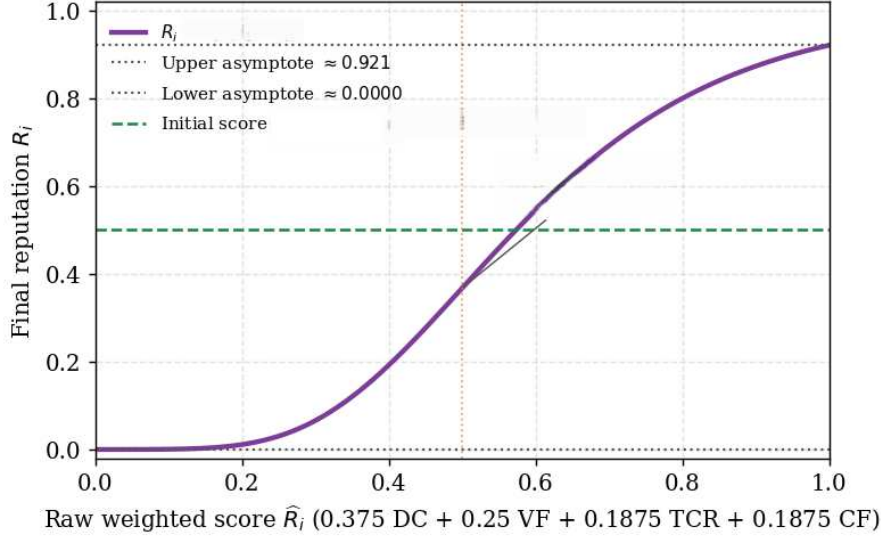


Fig. 10 Gompertz reputation mapping curve and asymptotic boundaries.

as $R_i \leftarrow R_i \cdot \exp(-\alpha \Delta t / \Delta t_{max})$, where Δt is the elapsed idle time and $\Delta t_{max} = 30$ rounds. Fig. 11 simulates an honest worker who is active for 40 rounds, reaching a mature score near 0.90, and then goes idle. Under a moderate decay rate ($\alpha = 2$), the score falls to 0.12 after one further inactivity window and below 0.01 after two, confirming that stale, unearned trust cannot persist indefinitely; milder ($\alpha = 1$) and sharper ($\alpha = 4$) settings bound the range of practical deployment choices.

The central security claim of the incentive layer is that weighting votes by reputation blunts Sybil and collusion attacks. Fig. 12 quantifies Sybil resistance: because every freshly created Sybil identity starts at the low initial score while established honest workers converge near 0.91, an attacker’s effective share of voting power is consistently below their raw headcount. Under naive one-vote-per-identity counting an attacker reaches the 50% tipping point with about 100 Sybils; under reputation-weighting the same 100 Sybils command only 30.6% of the voting power, and the 50% threshold is pushed out to roughly 235 identities. The attacker must therefore invest far more effort to influence the validation outcome.

Fig. 13 examines collusion within a validation committee, in which colluding validators all vote to accept deliberately bad data. With an unweighted majority rule the committee accepts the bad submission as soon as colluders reach 50% of the seats. Reputation-weighting raises this bar to roughly 59%: acceptance stays near 0% up to a 55% colluder fraction, crosses 50% acceptance around 59–60%, and only reaches full acceptance at 65%, because the remaining high-reputation honest validators still outweigh a numerical majority of lower-scored colluders. Forcing attackers to capture a supermajority of reputation, rather than a simple majority of seats, materially strengthens the Byzantine resilience of the validation step.

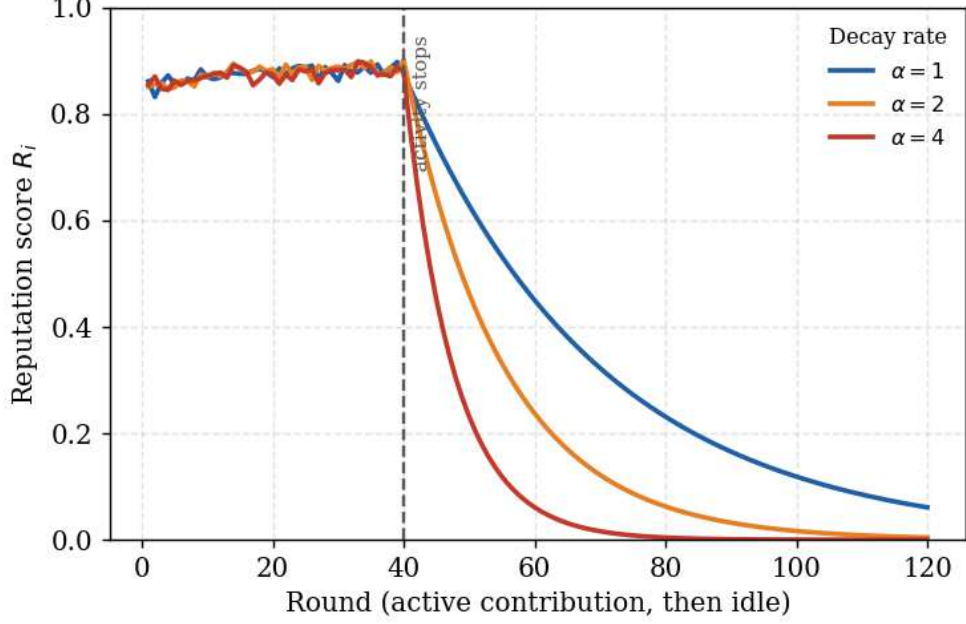


Fig. 11 Reputation decay during idle periods across decay rates ($\alpha \in \{1, 2, 4\}$).

5.7 End-to-End Latency Budget and Discussion

Combining the measured components yields the end-to-end cost of a single anonymous submission. The client first generates an authentication proof (≈ 1.40 s), which the USERMANAGER contract verifies in 2.75 ms; the encrypted payload is then pinned to IPFS and its CID, together with the reputation update, is committed through the ordering service. The critical path is therefore dominated by two factors that are both independent of the cryptographic verification: client-side proof generation and the orderer batch-cutting delay. Cryptographic verification, the part executed redundantly by every peer, contributes less than 10 ms. This validates the central design thesis of RBCrowd4RS: strong anonymity and on-chain verifiability can be achieved without imposing a prohibitive consensus-time penalty, since the expensive proving work is pushed to the client and the on-chain verifier remains succinct.

As presented in related work section, RBCrowd4RS is unlike reputation-server designs that retain a traceability backdoor to the real identity [19]. It achieves identity unlinkability via the Merkle-membership authentication proof, while still preserving accountability through the on-chain reputation ledger. Together with the constant-size, sub-3 ms on-chain verification, this places the system in a favorable region of the privacy/performance trade-off space that prior work treats as mutually exclusive.

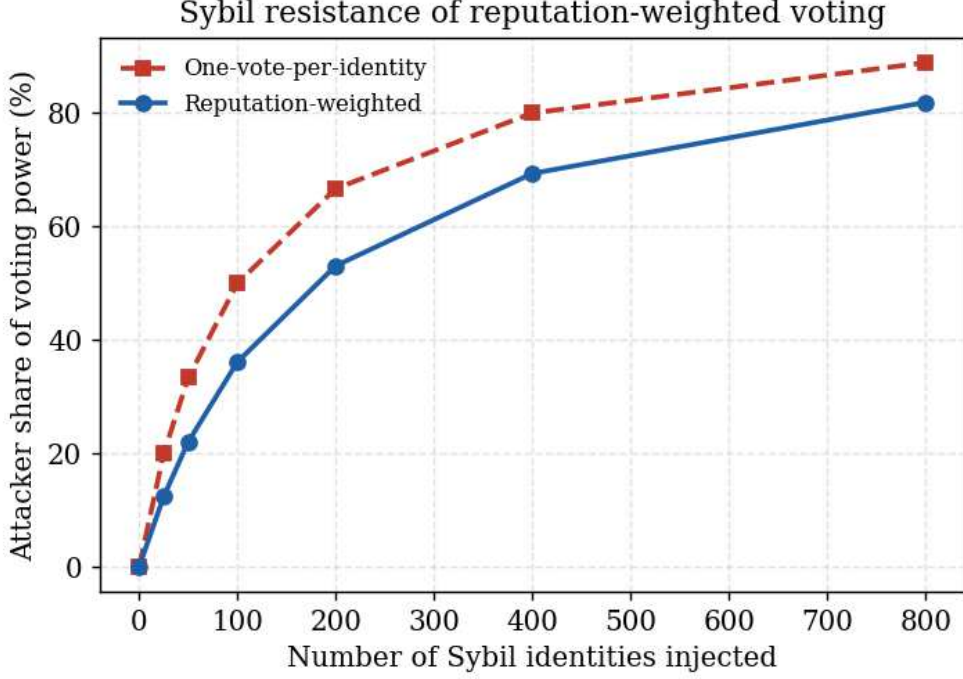


Fig. 12 Attacker voting power share under increasing Sybil identity injection.

6 Conclusion

This paper introduced RBCrowd4RS, a multi-layer, privacy-preserving, blockchain-based mobile-crowdsensing system that collects road data without forcing a choice between contributor anonymity and public accountability. The design integrates a consortium ledger and three smart contracts, with a PLONK zk-SNARK authentication circuit, and a dynamic reputation management model.

Prototype benchmarks via Hyperledger Caliper validate the framework’s efficiency and security. Authentication proofs remain a constant 520 byte and verify in 2.75ms, scaling beyond one million participants with $O(1)$ on-chain cost. Region-gated opt-in validation cuts median validator distance by $5.97\times$ while 79.0% of submissions reach the minimum validation quorum, confirming that identity unlinkability and real-time accountability can operate efficiently in practice. Finally, the dynamic reputation management model maintains a wide reputation margin, mitigates Sybil attacks, and raises Byzantine collusion tolerance from 50% to 59%. Because the critical path is dominated by client-side proof generation, strong unlinkability and on-chain verifiability are shown to be attainable in real time.

In future work, we plan to extend this framework in three principal directions. First, we will fine-tune the ShuffleNet v2 classification head on domain-specific road anomaly datasets (such as RDD2020/RDD2022) to evaluate on-device detection accuracy, precision, and false-rejection rates alongside inference latency. Second, to eliminate the

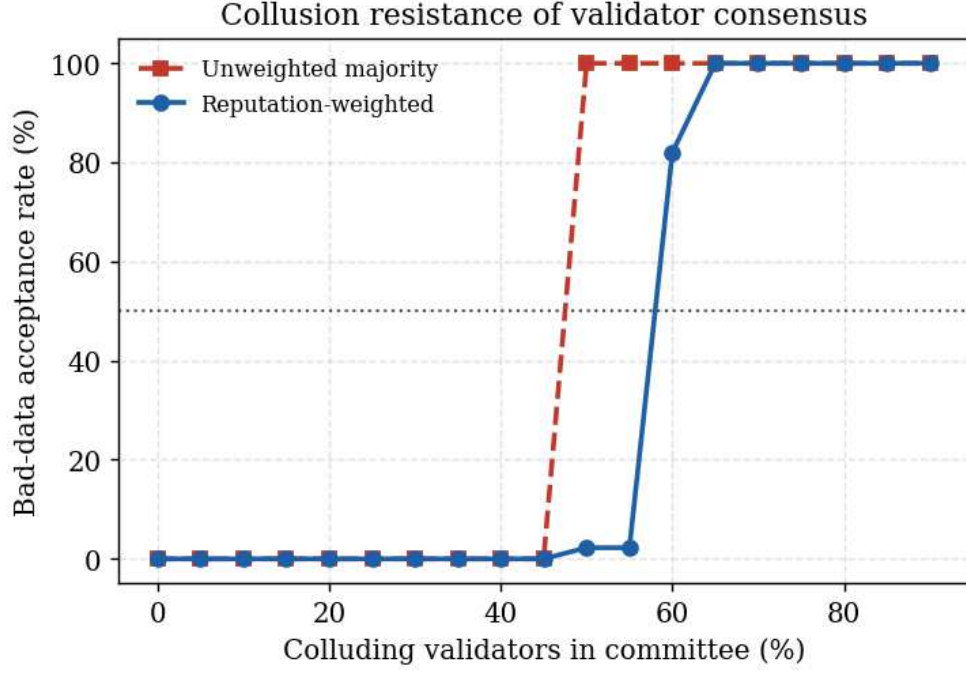


Fig. 13 Bad-data acceptance rate versus committee colluder percentage under unweighted and reputation-weighted consensus.

write-path bottleneck identified during load testing, we will implement composite-key state sharding to alleviate hot-key MVCC contention on the global leaderboard index. Finally, we intend to integrate our companion attribute- and role-based access control (A&R-BAC) scheme to deliver end-to-end, policy-governed data consumption for municipal transportation authorities and emergency services.

References

- [1] World Health Organization: Save LIVES: A Road Safety Technical Package. World Health Organization, Geneva, Switzerland (2017). <https://www.who.int/publications/i/item/save-lives-a-road-safety-technical-package>
- [2] Belin, M.-Å., Tillgren, P., Vedung, E.: Vision Zero – a road safety policy innovation. *International Journal of Injury Control and Safety Promotion* **19**(2), 171–179 (2012) <https://doi.org/10.1080/17457300.2011.635213>
- [3] Ali, N.A., Taha, A.-E.M., Barka, E.: Integrating blockchain and IoT/ITS for safer roads. *IEEE Network* **34**(1), 32–37 (2020) <https://doi.org/10.1109/MNET.001.1900146>

- [4] Maqour, Z., El Bakkali, H., Benhaddou, D., Benbrahim, H., Abou-Zbiba, W., El Gadi, H., Al-Fuqaha, A., Anan, M., Taha, A.-E.: An overview of funded research projects in the MENA region on intelligent transportation systems. In: 2023 International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 313–320. IEEE, Marrakesh, Morocco (2023). <https://doi.org/10.1109/IWCMC58020.2023.10182803>
- [5] Maqour, Z., El Bakkali, H., Benhaddou, D., Benbrahim, H.: Blockchain-based privacy preservation in mobile crowdsensing: A systematic literature review. *Computer Science Review* **61**, 100939 (2026) <https://doi.org/10.1016/j.cosrev.2026.100939>
- [6] El-Wakeel, A.S., Li, J., Noureldin, A., Hassanein, H.S., Zorba, N.: Towards a practical crowdsensing system for road surface conditions monitoring. *IEEE Internet of Things Journal* **5**(6), 4672–4685 (2018) <https://doi.org/10.1109/JIOT.2018.2807408>
- [7] Wang, X., Ning, Z., Hu, X., Ngai, E.C.-H., Wang, L., Hu, B., Kwok, R.Y.K.: A city-wide real-time traffic management system: Enabling crowdsensing in social Internet of Vehicles. *IEEE Communications Magazine* **56**(9), 19–25 (2018) <https://doi.org/10.1109/MCOM.2018.1701065>
- [8] Wang, L., Cao, Z., Zhou, P., Zhao, X.: Towards a smart privacy-preserving incentive mechanism for vehicular crowd sensing. *Security and Communication Networks* **2021**, 5580089 (2021) <https://doi.org/10.1155/2021/5580089>
- [9] Tian, Y., Zhou, T., Zhou, X., Zhong, W., Yang, X.: Attribute-based linear homomorphic signature scheme based on key policy for mobile crowdsensing. *Journal of King Saud University Computer and Information Sciences* **36**(6), 102114 (2024) <https://doi.org/10.1016/j.jksuci.2024.102114>
- [10] Huang, K.L., Kanhere, S.S., Hu, W.: Are you contributing trustworthy data? The case for a reputation system in participatory sensing. In: Proceedings of the 13th ACM International Conference on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM), pp. 14–22. ACM, Bodrum, Turkey (2010). <https://doi.org/10.1145/1868521.1868526>
- [11] Gabizon, A., Williamson, Z.J., Ciobotaru, O.: PLONK: Permutations over Lagrange-Bases for Oecumenical Noninteractive Arguments of Knowledge. *Cryptography ePrint Archive*, Paper 2019/953 (2019). <https://eprint.iacr.org/2019/953>
- [12] Maqour, Z., El Bakkali, H., Benhaddou, D., Benbrahim, H.: RBCrowd: A reliable blockchain-based reputation management framework for privacy preservation in mobile crowdsensing. *Future Internet* **18**(1), 65 (2026) <https://doi.org/10.3390/fi18010065>

- [13] Ma, N., Zhang, X., Zheng, H.-T., Sun, J.: ShuffleNet V2: Practical guidelines for efficient CNN architecture design. In: Computer Vision – ECCV 2018. Lecture Notes in Computer Science, pp. 122–138. Springer, Cham, Switzerland (2018). https://doi.org/10.1007/978-3-030-01264-9_8
- [14] Ju, R.-Y., Lin, T.-Y., Jian, J.-H., Chiang, J.-S., Yang, W.-B.: ThreshNet: An efficient DenseNet using threshold mechanism to reduce connections. IEEE Access **10**, 82834–82843 (2022) <https://doi.org/10.1109/ACCESS.2022.3196492>
- [15] Wang, F.-Y., *et al.*: Transportation 5.0: The DAO to safe, secure, and sustainable intelligent transportation systems. IEEE Transactions on Intelligent Transportation Systems **24**(10), 10262–10278 (2023) <https://doi.org/10.1109/TITS.2023.3305380>
- [16] Sharma, A., Awasthi, Y., Kumar, S.: The role of blockchain, AI and IoT for smart road traffic management system. In: Proceedings of the IEEE India Council International Subsections Conference (INDISCON), pp. 289–296. IEEE, Visakhapatnam, India (2020). <https://doi.org/10.1109/INDISCON50162.2020.00065>
- [17] Kang, J., Xiong, Z., Niyato, D., Ye, D., Kim, D.I., Zhao, J.: Toward secure blockchain-enabled Internet of Vehicles: Optimizing consensus management using reputation and contract theory. IEEE Transactions on Vehicular Technology **68**(3), 2906–2920 (2019) <https://doi.org/10.1109/TVT.2019.2894944>
- [18] Javed, F., Mangues-Bafalluy, J., Zeydan, E., Blanco, L.: Trustworthy reputation for federated learning in O-RAN using blockchain and smart contracts. IEEE Open Journal of the Communications Society **6**, 1343–1362 (2025) <https://doi.org/10.1109/OJCOMS.2025.3540159>
- [19] Mun, H., Han, K., Yeun, H.K., Damiani, E., Puthal, D., Kim, T.-Y., Yeun, C.Y.: Emerging blockchain and reputation management in federated learning: Enhanced security and reliability for Internet of Vehicles (IoV). IEEE Transactions on Vehicular Technology **74**(2), 1893–1908 (2025) <https://doi.org/10.1109/TVT.2024.3456852>
- [20] Capponi, A., Fiandrino, C., Kantarci, B., Foschini, L., Kliazovich, D., Bouvry, P.: A survey on mobile crowdsensing systems: Challenges, solutions, and opportunities. IEEE Communications Surveys & Tutorials **21**(3), 2419–2465 (2019) <https://doi.org/10.1109/COMST.2019.2914030>
- [21] Sun, X., Yu, F.R., Zhang, P., Sun, Z., Xie, W., Peng, X.: A survey on zero-knowledge proof in blockchain. IEEE Network **35**(4), 198–205 (2021) <https://doi.org/10.1109/MNET.011.2000473>

- [22] Zhang, Y., Tu, Z., Wang, X., Yuan, Y., Wang, Z., Chen, Y.: Toward privacy-accountable authentication in vehicular networks: An HQC-based context-aware linkable ring signature scheme. *Journal of King Saud University Computer and Information Sciences* **38**(6), 616 (2026) <https://doi.org/10.1007/s44443-026-00993-z>
- [23] Benet, J.: IPFS – Content Addressed, Versioned, P2P File System. arXiv:1407.3561 (2014). <https://arxiv.org/abs/1407.3561>
- [24] Androulaki, E., *et al.*: Hyperledger Fabric: A distributed operating system for permissioned blockchains. In: *Proceedings of the 13th EuroSys Conference*, pp. 1–15. ACM, Porto, Portugal (2018). <https://doi.org/10.1145/3190508.3190538>
- [25] Botrel, G., Piellard, T., El Housni, Y., Kubjas, I., Tabaie, A.: ConsenSys/gnark: A Fast zk-SNARK Library. Zenodo (2023). <https://doi.org/10.5281/zenodo.5819104> . <https://github.com/ConsenSys/gnark>
- [26] Albrecht, M., Grassi, L., Rechberger, C., Roy, A., Tiessen, T.: MiMC: Efficient encryption and cryptographic hashing with minimal multiplicative complexity. In: *Advances in Cryptology – ASIACRYPT 2016. Lecture Notes in Computer Science*, vol. 10031, pp. 191–219. Springer, Berlin, Heidelberg (2016). https://doi.org/10.1007/978-3-662-53887-6_7
- [27] Hyperledger Foundation: Hyperledger Caliper: A Blockchain Performance Benchmark Framework. <https://hyperledger.github.io/caliper/> (2023)
- [28] Huang, Y., Lu, Y., Fan, L., Xiang, K., Ma, H.: Fault arc detection method based on improved ShuffleNet V2 network. *Processes* **13**(1), 135 (2025) <https://doi.org/10.3390/pr13010135>