Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

# Sovereign-Mohawk: Machine-Checked Hierarchical Byzantine Resilience and Probabilistic Composition for Federated Learning

**Ryan Williams**

r.williamspbg@gmail.com

Self https://orcid.org/0009-0005-1126-7004

**Research Article**

**Keywords:** Federated Learning, Byzantine Fault Tolerance, Formal Verification, Lean 4, Hierarchical Aggregation, Differential Privacy, Zero-Knowledge Proofs

**Posted Date:** November 12th, 2026

**DOI:** https://doi.org/10.21203/rs.3.rs-11694611/v1

**License:** This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

**Additional Declarations:** The authors declare no competing interests.

# Sovereign-Mohawk: Machine-Checked Hierarchical Byzantine Resilience and Probabilistic Composition for Federated Learning

**Abstract**

Federated learning (FL) at scale requires aggregating model updates across multi-tier topologies while tolerating adversarial participants. A common assumption in hierarchical FL is that if every local aggregation tier enforces an honest-majority guard, the composition inherits global Byzantine resilience. In this paper, we present **Sovereign-Mohawk**, an FL architecture and Lean 4 formal development that refutes and repairs this assumption.

First, we present a machine-checked negative result: a concrete counterexample in Lean 4 showing that naive deterministic composition of per-tier Multi-Krum guards fails, allowing a 60% global Byzantine leaf fraction to pass local filters under concentrated placement. Second, we formalize a probabilistic repair proving that under explicit $i.i.d.$ Bernoulli committee sampling, the probability of any tier exceeding its Byzantine filtering threshold is strictly bounded by a Chernoff bound over rationals, verified for $T = 200$ committees and $c = 50{,}000$ leaves per committee via `native_decide` ($P_{\text{fail}} \leq 200 \times 2^{-1000}$). Third, we couple these formal artifacts with a deployable Go runtime featuring Rényi differential privacy, flat Groth16/BN254 zero-knowledge verifiability, and post-quantum key exchange (X25519-ML-KEM-768). Finally, we evaluate the system across multi-node Kind (Kubernetes-in-Docker) clusters on single hosts (up to 500 pods), CI swarm matrices (500–1,500 nodes), and a 1,000-node simulator target under safe (10% Byzantine) and edge (35% Byzantine) adversarial regimes, demonstrating microsecond aggregation overhead (30.63 $\mu$s) and minimal accuracy impact (1.8% gap vs. clean baseline). Every formal claim is bound to implementation code via a publicly audited traceability matrix enforced in continuous integration.

Federated Learning, Byzantine Fault Tolerance, Formal Verification, Lean 4, Hierarchical Aggregation, Differential Privacy, Zero-Knowledge Proofs.

# Introduction

Federated learning (FL) enables decentralized entities to collaboratively train machine learning models without centralizing raw datasets . As FL scales to global deployment scenarios involving millions of edge devices, system designers face what we term the *Trust-Scale Paradox*: larger node populations increase the probability of encountering Byzantine failures, intentional model poisoning attacks, and network partitioning, while the overhead of cryptographic and statistical verification must remain strictly sub-linear in total participant count.

To manage bandwidth and processing overhead, state-of-the-art FL deployments organize participants into multi-tier topological hierarchies (e.g., Edge → Regional → Continental → Global) . A tempting assumption in hierarchical FL design is *compositional safety*: if every individual aggregation node enforces a localized Byzantine filter (such as Multi-Krum ), the overall network retains global Byzantine resilience up to the per-tier tolerance threshold.

In this work, we demonstrate that this assumption is fundamentally flawed, and we present a fully machine-checked framework in Lean 4 that mathematically refutes deterministic hierarchical composition while constructing a machine-checked probabilistic repair. We embody these insights in **Sovereign-Mohawk**, a high-assurance FL system coupling a production Go runtime with an automated formal traceability pipeline.

## Core Contributions

This paper makes four main contributions centered on formal Byzantine analysis, system design, and empirical evaluation:

1. **Machine-Checked Counterexample (Negative Result):** We formalize a concrete two-level hierarchy in Lean 4 (`Theorem1BFT.lean`) where local Multi-Krum guards ($2f < n$) pass at every intermediate aggregation tier, yet the true global Byzantine participant ratio reaches $60\%$. This disproves naive deterministic hierarchical composition permanently.
2. **Chernoff-Based Probabilistic Repair:** We prove in Lean 4 that under explicit $i.i.d.$ Bernoulli committee assignment, global resilience is restored probabilistically. We cleanly separate the general KL-divergence expression from the machine-checked rational bound, verifying in Lean using `native_decide` that for $T = 200$ committees of size $c = 50{,}000$ under $10\%$ global adversary concentration, the failure probability is upper-bounded by $200 \times 2^{-1000}$.
3. **High-Assurance Runtime Integration:** We integrate the repaired hierarchical aggregation scheme into a runnable Go runtime equipped with Rényi Differential Privacy (RDP) accounting, flat Groth16/BN254 zero-knowledge proof binding over MiMC commitments, and hybrid X25519-ML-KEM-768 key exchange.
4. **Grounded System Evaluation:** We evaluate Sovereign-Mohawk across single-host multi-node Kubernetes (Kind) clusters and CI swarm runtime environments up to 1,500 active containerized nodes under aggressive adversarial regimes ($10\%$ to $35\%$ Byzantine workers), measuring accuracy preservation, latency overheads, ablation profiles, and verification throughput.

## Scientific Scope & Honesty Statement

In accordance with rigorous formal methods practices, we explicitly bound our claims. Earlier internal drafts modeled parameter bounds up to $10^7$ nodes ($b = 10, h = 7$). In this paper, $10^7$ is recognized strictly as a *topological design capacity formula*, whereas empirical scale measurements are reported strictly up to 500 single-host Kubernetes pods and 1,500 simulated container nodes. Furthermore, Gaussian noise bounds for RDP and abstract Groth16 proof costs are explicitly identified where they rely on Lean axioms versus fully unwound definitions. Every claim in this paper maps to an entry in our public traceability matrix (`FORMAL_TRACEABILITY_MATRIX.md`).

# System Architecture and Threat Model

## Four-Tier Hierarchy

Sovereign-Mohawk structures participant nodes into a four-tier tree topology with branching factor $b = 10$:

- **Tier 1 (Edge):** Devices compute local stochastic gradient descent (SGD) updates and apply local differential privacy noise. Synchrony bound $\Delta_1 = 30$s.
- **Tier 2 (Regional):** Aggregates edge updates using localized Multi-Krum filtering over homomorphically encrypted (Paillier 2048-bit) or plaintext vector slices. Synchrony bound $\Delta_2 = 150$s.
- **Tier 3 (Continental):** Performs cross-regional aggregation, verifying zero-knowledge validity proofs generated by Tier 2 aggregators. Synchrony bound $\Delta_3 = 600$s.
- **Tier 4 (Global Orchestrator):** Finalizes the global model update and updates the state ledger.

## Network and Adversary Model

We adopt the partially synchronous network model of Dwork, Lynch, and Stockmeyer , assuming a Global Stabilization Time (GST) after which message delivery delays are bounded by $\Delta$.

An $(f, d)$-*Byzantine Adversary* controls up to $f$ participant nodes and may delay up to $d$ honest messages per tier by at most $\Delta_{\text{adv}}$. Byzantine nodes are assumed to possess full knowledge of the global model state and may collude across arbitrary tiers. Byzantine nodes may inject arbitrary state vectors, send conflicting updates to different aggregators (equivocation), or remain silent (straggler attack). Cryptographic primitives (Groth16/BN254, XMSS signatures, ML-KEM-768) are assumed computationally unforgeable under standard cryptographic hardness assumptions.

# Formal Analysis: Disproof and Repair

This section details the primary theoretical contributions machine-checked inside Lean 4.

## Single-Tier Byzantine Guard

At a single aggregation tier with $n$ submitted gradient vectors, the Multi-Krum algorithm selects a subset of vectors that minimize pairwise Euclidean distances. Algorithm [alg:multikrum] outlines the local filtering process.

$m \leftarrow n - f - 2$ $S \leftarrow$ indices of the $m$ smallest scores $s_i$

In Lean 4 (`Theorem1BFT.lean`), we formalize the single-tier honest-majority guard condition:

$$\mathcal{G}_{\text{local}}(n, f) \equiv (2f < n) \wedge (9f < 5n)$$

When $\mathcal{G}_{\text{local}}$ holds, Multi-Krum guarantees that the aggregated vector lies within a bounded hyper-sphere centered at the true mean of honest gradients.

## Disproof of Deterministic Composition

It is widely conjectured that if $\mathcal{G}_{\text{local}}$ holds at every regional committee in a hierarchy, the overall leaf population satisfies a global Byzantine bound $f_{\text{global}}/n_{\text{global}} < 1/2$. We disprove this conjecture in Lean 4 via a machine-checked counterexample that matches the repository's exact formal statement.

**Counterexample 1** (Composition Disproof in Lean 4). *Consider a two-level hierarchy with $T = 2$ regional aggregators $(A_1, A_2)$. Let $A_1$ manage $c_1 = 100$ leaves, and $A_2$ manage $c_2 = 100$ leaves, yielding $N = 200$ total leaves.*

- *Aggregator $A_1$ contains $f_1 = 49$ Byzantine nodes and $51$ honest nodes.*
- *Aggregator $A_2$ contains $f_2 = 71$ Byzantine nodes and $29$ honest nodes.*

*Aggregator $A_1$ satisfies $2f_1 < c_1$ ($98 < 100$), so its local guard passes. However, if the root orchestrator weighs $A_1$ and $A_2$ based on passed filter outputs, an adversary placing 120 total Byzantine nodes across $N = 200$ leaves achieves a $60\%$ global Byzantine fraction ($120/200 = 0.60$), violating global honest majority while intermediate guards report compliance.*

The Lean 4 formalization verifies this construction explicitly:

```
theorem hierarchical_composition_counterexample :
  exists (tree : TierTree), (forall c in tree.committees,
    LocalGuard c) and (GlobalByzantineFraction tree > 3 / 5) := by ...
```

This machine-checked result establishes that local guard filtering does not form a sub-category under deterministic composition.

## Chernoff-Based Probabilistic Repair

To restore system-wide guarantees, we formalize a probabilistic repair mechanism. Instead of deterministic placement, nodes are assigned to $T$ committees of capacity $c$ via an $i.i.d.$ Bernoulli process with global Byzantine probability $p = P(\text{Byzantine})$.

First, we state the continuous analytical Chernoff bound:

$$P\left(\sum_{i=1}^{c} X_i \geq k\right) \leq \exp\left(-c \cdot D_{\text{KL}}\left(\frac{k}{c} \parallel p\right)\right)$$

Second, to avoid floating-point non-determinism inside the proof assistant, we formalize a discrete rational arithmetic ($\mathbb{Q}$) version of the Chernoff bound at base $z = 2$ in `Theorem1BFT.lean`.

**Theorem 1** (Rational Chernoff Bound Instance). *For parameters $T = 200$ committees, committee capacity $c = 50{,}000$, global adversary proportion $p = 0.10$, and filter capacity threshold $k = 25{,}000$ (50%), the union bound over all committees satisfies: $P_{\text{fail}} \leq T \cdot \text{ChernoffRat}(c, p, k) \leq 200 \times 2^{-1000} \approx 1.87 \times 10^{-299}$ This bound is certified directly in Lean 4 via `native_decide` over 5,000-digit exact rational numbers.*

This machine-checked result proves that under uniform random assignment, hierarchical composition is overwhelmingly secure against concentrated Byzantine placement.

# Implementation & Formal Refinement

The Sovereign-Mohawk architecture bridges high-level Lean specifications with an industrial Go runtime through verified refinement relations and continuous integration gates.

*Sovereign-Mohawk System Architecture and Formal Refinement Pipeline.*

## Rényi Differential Privacy (RDP) Accountant

Privacy is enforced using Rényi Differential Privacy . Each local update applies Gaussian noise $\sigma^2$ calibrated to fixed order $\alpha = 10$. The runtime accountant tracks additive composition across training rounds. To convert accumulated RDP bounds to standard $(\epsilon, \delta)$-DP, we apply the transformation:

$$\epsilon(\delta) = \epsilon_{\text{rdp}} + \frac{\ln(1/\delta)}{\alpha - 1}$$

Because transcendental functions like $\ln(z)$ are non-computable in constructivist logic, we formalize a two-sided rational sandwich bound in `Refinement/RDPLogBound.lean`:

$$\text{epsLowerRat}(z) \leq \ln(z) \leq \text{epsUpperRat}(z) \quad \forall z \in \mathbb{Q}^+$$

This allows Lean's kernel to compute exact bounding intervals for $\epsilon(\delta)$ at target values $\delta \in \{10^{-3}, 10^{-5}, 10^{-8}\}$. In CI, a Python trace validator (`check_refinement.py`) replays production Go execution logs against Lean specifications to detect specification drift.

## Zero-Knowledge Proof Verification

To prevent corrupt regional aggregators from tampering with intermediate vector sums, aggregators submit flat Groth16 proofs over the BN254 elliptic curve . The circuit enforces that the output vector matches the MiMC hash commitment of valid input gradients.

We deliberately avoid claims of recursive proof composition (e.g., Halo2 or Pickles cycles), as BN254 does not form a pairing-friendly curve cycle. The Go runtime lazily initializes the Groth16 verifying key via `sync.Once` to eliminate startup latency overheads.

## Post-Quantum Cryptographic Stack

Sovereign-Mohawk incorporates post-quantum cryptographic controls by default:

- **Transport Layer:** Hybrid X25519-ML-KEM-768 key exchange for inter-node TLS channels.
- **Attestation Identity:** XMSS stateful hash-based signatures bound to TPM hardware identities.
- **Ledger Migration:** Epoch-enforced dual-signature transitions (enforced target date: 2027-12-31), requiring both classical ECDSA and PQC signatures for state updates.

## Formal Traceability Matrix

Every formal claim in the repository is logged in a machine-readable matrix (`FORMAL_TRACEABILITY_MATRIX.md`). Table [tab:traceability] summarizes key inventory items.

# Experimental Evaluation & Ablations

We evaluate Sovereign-Mohawk across single-host multi-node Kind (Kubernetes-in-Docker) environments, CI swarm matrices, and simulated runtime harnesses up to 1,500 active containerized nodes. Our public artifacts and logs are committed directly in `captured_artifacts/500node_scale_Proof`. Our evaluation targets four questions: 1. How does the probabilistic repair perform under active Byzantine attacks across distributed pods? 2. What accuracy trade-off is introduced by localized Multi-Krum aggregation and differential privacy noise? 3. What is the execution latency and throughput cost of zero-knowledge and cryptographic controls? 4. How effectively does the Go runtime scale under local containerized environments?

## Experimental Setup

Experiments were executed on a dedicated workstation featuring an AMD EPYC 7763 processor (64 cores, 256GB RAM) running Ubuntu 22.04 with Linux 6.5. Pod networking and container resource allocations were enforced via Kind (multi-node Docker container nodes running on a single host) and Docker Compose harnesses. We test under two adversary profiles:

- **Safe Profile:** 10% global Byzantine nodes applying sign-flip attacks.
- **Edge Profile:** 35% global Byzantine nodes executing coordinated state manipulation attacks.

## Byzantine Resilience & Model Convergence

Figure 2 tracks model convergence on CIFAR-10 across 100 training rounds under Safe and Edge profiles with and without Multi-Krum filtering.

*Model Test Accuracy under Safe (10%) and Edge (35%) Byzantine Attack Profiles across Containerized Swarm Nodes.*

Under the Edge profile (35% Byzantine), unmitigated FedAvg diverges rapidly within 10 training rounds. In contrast, Sovereign-Mohawk's hierarchical Multi-Krum aggregator maintains a steady convergence curve, reaching 86.4% accuracy after 100 rounds—representing a small 1.8% gap compared to the unperturbed clean baseline (88.2%).

## Ablation Analysis

Table 1 provides an ablation breakdown isolating the impact of each core security component on accuracy and round runtime overhead on a 500-pod setup.

*System Component Ablation Matrix (500 Pods, Round 50)*

| Configuration / Feature Vector | CIFAR-10 Acc (%) | Round Delay |
|---|---|---|
| Standard FedAvg (No Security / No Attack) | 88.2% | 110 ms |
| + Multi-Krum Guard Only (35% Attack) | 87.8% | 124 ms |
| + Multi-Krum + RDP Privacy ($\epsilon = 2.0$) | 86.4% | 128 ms |
| + Full Stack (Groth16 + KEM-768 + RDP) | 86.4% | 145 ms |

The ablation study confirms that differential privacy noise accounts for most of the 1.8% accuracy differential, while Groth16 proof verification and post-quantum key transport introduce manageable runtime overhead (17 ms per round).

## Containerized Swarm Scale Benchmarks

We evaluate round completion latency across scaled node counts (500 to 1,500 containerized instances). Table 2 presents latency breakdowns across aggregation tiers.

*Containerized Swarm Benchmarks (Mean Latency per Round)*

| Active Nodes | Tier 1 (Edge) | Tier 2 (Reg.) | Tier 3 (Cont.) | Total Round |
|---|---|---|---|---|
| 500 | 12 ms | 45 ms | 88 ms | 145 ms |
| 1,000 | 24 ms | 82 ms | 140 ms | 246 ms |
| 1,500 | 38 ms | 125 ms | 210 ms | 373 ms |

Across 1,500 active containerized nodes, total round completion latency remains under 400 ms (373 ms), confirming that multi-tier aggregation introduces minimal processing overhead.

## Microbenchmarks

Table [tab:microbench] provides single-host microbenchmark timings for core runtime operations measured on a single AMD EPYC core.

The Groth16 verification throughput of 94.8 ops/s per core allows a regional aggregator to process validity proofs from nearly 100 sub-nodes per second per dedicated core.

# Related Work

**Byzantine-Robust FL:** Blanchard et al. introduced Multi-Krum for single-tier robustness. Subsequent schemes such as BNS and Krum variants explored geometric medians . However, these works focused on flat network topologies and did not model compositional failure modes under hierarchical aggregation.

**Hierarchical Federated Learning:** Hierarchical FL topologies have been proposed to optimize network bandwidth in edge environments . Existing systems rely on heuristic assumptions for security rather than machine-checked composition proofs.

**Formal Verification in FL:** Formal verification of distributed consensus systems (e.g., Raft, Paxos) using Ivy, Coq, or Lean is well studied . However, formalizing statistical ML properties—specifically Byzantine filtering thresholds and differential privacy composition—remains rare. Recent work by He and Zhang explored probabilistic bounds but did not provide machine-checked proofs or refutations.

**Verifiable Computation in FL:** Systems combining zero-knowledge proofs with FL (e.g., ZK-FL) frequently rely on heavy recursive SNARK constructions . Sovereign-Mohawk adopts flat Groth16 proofs over MiMC commitments to avoid the high overheads and complex curve cycles associated with recursive SNARKs.

# Limitations & Open Gaps

We explicitly log remaining technical limitations:

1. **Axiomatized Measure Theory:** Lean 4 bounds for Gaussian mechanism RDP are defined as approved axioms rather than fully unwound Mathlib measure theory proofs.
2. **Trusted Setup:** Groth16 verification requires an initial trusted setup phase. A future transition to Halo2 or STARK-based schemes would eliminate structured reference string dependencies.
3. **Sampling Assumptions:** The probabilistic repair relies strictly on $i.i.d.$ Bernoulli assignment. Non-uniform or correlated adversarial node placement remains an open research topic.

4. **Empirical Multi-Host Scaling:** While single-host multi-node Kind container benchmarks up to 1,500 nodes demonstrate high runtime efficiency, physical multi-host and multi-region WAN deployments remain future empirical work.

## Conclusion

Sovereign-Mohawk reframes hierarchical federated learning by exposing a fundamental vulnerability in naive local guard composition and providing a machine-checked probabilistic solution. By pairing a Lean 4 formal development with a high-performance Go runtime, we demonstrate that formal verification can directly inform system design in large-scale ML systems. All formal developments, microbenchmarks, and continuous integration workflows are publicly available.

## Acknowledgments

We thank the maintainers of Lean 4, Mathlib, the gnark zero-knowledge library, and the Go open-source ecosystem.

99

H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proc. AISTATS*, 2017.

P. Blanchard, E. M. El Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. NeurIPS*, 2017.

C. Dwork, N. Lynch, and L. Stockmeyer, "Consensus in the presence of partial synchrony," *Journal of the ACM*, vol. 35, no. 2, pp. 288–323, 1988.

I. Mironov, "Rényi differential privacy," in *Proc. IEEE Computer Security Foundations Symposium (CSF)*, 2017.

J. Groth, "On the size of pairing-based non-interactive arguments," in *Proc. EUROCRYPT*, pp. 305–326, 2016.

Y. Chen, L. Su, and J. Xu, "Distributed statistical machine learning in adversarial settings: Byzantine gradient descent," *ACM Measurement and Analysis of Computing Systems*, vol. 1, no. 2, 2017.

D. Yin, Y. Chen, R. Kannan, and P. Bartlett, "Byzantine-robust distributed learning: Towards optimal statistical rates," in *Proc. ICML*, 2018.

L. Liu, J. Zhang, S. Song, and K. B. Letaief, "Client-edge-cloud hierarchical federated learning," *IEEE Journal on Selected Areas in Communications*, vol. 38, no. 10, 2020.

M. S. H. Abad, E. Ozfatura, D. Gündüz, and O. Ercetin, "Hierarchical federated learning across heterogeneous wireless networks," *IEEE Transactions on Wireless Communications*, 2020.

C. Hawblitzel, J. Howell, J. Lorch, A. Narayan, B. Parno, D. Zhang, and B. Zill, "IronFleet: Proving practical distributed system implementations correct," in *Proc. SOSP*, 2015.

D. Woos, J. R. Wilcox, S. Antonov, T. Tatsiramos, M. D. Ernst, and T. Anderson, "Planning for change in a formal verification of Raft," in *Proc. OSDI*, 2016.

Y. He and J. Zhang, "Probabilistic Byzantine resilience for federated learning," *IEEE Transactions on Dependable and Secure Computing*, 2026.

E. Ben-Sasson, A. Chiesa, E. Tromer, and M. Virza, "Recursive composition and bootstrapping for SNARKs and proof-carrying data," in *Proc. STOC*, 2014.

T. Ryffel et al., "A generic framework for privacy preserving deep learning," *NeurIPS Privacy in Machine Learning Workshop*, 2018.

C. Dwork and A. Roth, “The algorithmic foundations of differential privacy,” *Foundations and Trends in Theoretical Computer Science*, vol. 9, no. 3–4, pp. 211–407, 2014.

D. J. Bernstein, “Curve25519: new Diffie-Hellman speed records,” in *Proc. Public Key Cryptography (PKC)*, 2006.

National Institute of Standards and Technology (NIST), “FIPS 203: Module-Lattice-Based Key-Encapsulation Mechanism Standard,” 2024.

R. Williams, “Sovereign Mohawk Proto: Formally Verified Hierarchical Federated Learning Engine,” GitHub Repository, 2026. https://github.com/rwilliamspbg-ops/Sovereign-Mohawk-Proto.