zkFL-Health: Blockchain-Enabled Zero-Knowledge Federated Learning for Medical AI Privacy

Savvy Sharma
School of Arts And Technology
George Brown Polytechnic
Toronto, ON, Canada
osive.savvy@gmail.com

George Petrovic
School of Arts And Technology
Head of Blockchain Programme
George Brown Polytechnic
Toronto, ON, Canada
Djordje.Petrovic@georgebrown.ca

Sarthak Kaushik
School of Arts And Technology
George Brown Polytechnic
Toronto, ON, Canada
saki.osive@gmail.com

Abstract—Healthcare AI needs large, diverse datasets, yet strict privacy and governance constraints prevent raw data sharing across institutions. Federated learning (FL) mitigates this by training where data reside and exchanging only model updates, but practical deployments still face two core risks: (1) *privacy leakage* via gradients or updates (membership inference, gradient inversion) and (2) *trust in the aggregator*, a single point of failure that can drop, alter, or inject contributions undetected. We present zkFL-Health, an architecture that combines FL with zero-knowledge proofs (ZKPs) and Trusted Execution Environments (TEEs) to deliver privacy-preserving, *verifiably correct* collaborative training for medical AI. Clients locally train and commit their updates; the aggregator operates within a TEE to compute the global update and produces a succinct ZK proof (via Halo2/Nova) that it used exactly the committed inputs and the correct aggregation rule, without revealing any client update to the host. Verifier nodes validate the proof and record cryptographic commitments on-chain, providing an immutable audit trail and removing the need to trust any single party. We outline system and threat models tailored to healthcare, the zkFL-Health protocol, security/privacy guarantees, and a performance evaluation plan spanning accuracy, privacy risk, latency, and cost. This framework enables multi-institutional medical AI with strong confidentiality, integrity, and auditability, key properties for clinical adoption and regulatory compliance.

Index Terms—Federated learning, zero-knowledge proofs, blockchain, medical AI, privacy, verifiability, compliance.

I. INTRODUCTION

Data–driven healthcare promises earlier diagnosis, equitable triage, and personalized therapies, but clinical data are siloed across institutions under stringent privacy and governance regimes. *Federated learning* (FL) helps by training where data reside and sharing only model parameters [1]–[4]. In practice, however, two unresolved gaps prevent broad clinical deployment: (i) **privacy leakage** from model updates (e.g., membership inference and gradient inversion) [5]–[7] and (ii) **trust in the aggregator**, a single point of failure capable of dropping, altering, or injecting contributions without detection [8], [9]. Beyond privacy and integrity, hospitals and regulators increasingly require a *verifiable audit trail* that demonstrates how a model was produced [10], [11].

We propose **zkFL-Health**, a verifiable, privacy-preserving FL framework tailored for cross-silo medical AI. The core idea is to make the aggregator *provably honest*. After collecting client updates, the aggregator computes the global update and produces a succinct zero-knowledge (ZK) proof that it used exactly the committed inputs and the prescribed aggregation rule, without revealing any client's update [12]–[15]. A blockchain (permissioned consortium or public with appropriate data minimization) acts as the decentralized verifier and immutable log: verifier nodes check the proof and record cryptographic commitments on-chain, removing the need to trust any single party and enabling ex-post auditability [8], [10], [16].

Why FL alone is not enough. Keeping raw data local mitigates direct disclosure, but does not preclude leakage through gradients or weights [5]–[7]. Practical deployments also struggle to evidence that an aggregator neither excluded certain sites nor tampered with contributions.

The Limits of Existing PETs. Traditional privacy-enhancing technologies each address only parts of the problem:

- *Differential Privacy (DP)* protects against inference but degrades model utility (accuracy) [17].
- *Secure Aggregation (SecAgg)* protects confidentiality but lacks auditability; a poisoned global model cannot be traced back to the source.
- *Trusted Execution Environments (TEEs)* protect confidentiality by isolating computation, but they function as opaque "black boxes" to the public. They do not natively provide an on-chain, publicly verifiable proof of correctness without relying on the hardware manufacturer's central trust authority [8], [18].

Our Hybrid Approach. We leverage TEEs strictly for *confidentiality* (hiding raw updates from the aggregator) while using Zero-Knowledge Proofs for *verifiability* (proving correctness to the public). This allows us to achieve high privacy without sacrificing the public audit trail.

Design principles. zkFL-Health follows: (1) *privacy-by-design* (no raw data leaves a site; only commitments/proofs go on-chain), (2) *verifiability-by-default* (every accepted global update is backed by a ZK proof), (3) *minimal on-chain footprint* (hashes, commitments, and proofs; models remain off-chain), (4) *algorithm agnosticism* (works with CNNs, transformers, and tabular models), and (5) *extensibility* (clean interfaces to layer DP, robust aggregation, or TEEs).

Feature	Vanilla FL	FL+SecAgg	FL+DP	zkFL-Health
Data Privacy	Low	High	High	**High (TEE)**
Verifiability	None	Low	None	**High (ZK)**
Trust Model	Centralized	Semi-Honest	Centralized	**Trustless**
Audit Trail	No	No	No	**Yes**
Utility Loss	None	None	High	**Minimal**

Scope and threat model (preview). We target cross-silo FL among hospitals and labs (tens of clients, heterogeneous data, regulated networks) [2]. The aggregator is untrusted (may deviate arbitrarily); clients are semi-honest by default, with provisions for malicious behavior. Adversaries may attempt update tampering, client omission, replay, Sybil injection, and inference from released models. zkFL-Health provides cryptographic integrity for aggregation and a verifiable process log; protections against model-level privacy attacks (e.g., membership inference) can be layered via DP without changing the verification flow [17].

Contributions. This paper makes the following contributions:

- A healthcare-oriented *system and threat model* for verifiable, privacy-preserving cross-silo FL using a hybrid ZK-TEE architecture [2], [8].
- The *zkFL-Health protocol*: client commitments and signatures, aggregator-side succinct ZK proofs of correct aggregation (using Halo2/Nova), and blockchain-backed verification/logging with strict data minimization [10], [12]–[14], [16].
- A *security, privacy, and compliance* analysis articulating confidentiality, integrity, auditability, and deployment considerations under HIPAA/GDPR mindsets.
- *Engineering guidance* for performance and scalability (hierarchical aggregation, recursive proofs, asynchronous rounds) and an evaluation methodology covering utility, privacy risk, latency, and cost [8], [15].

Practicality. Proof verification is lightweight; proof generation cost scales with model size and client count but is amenable to batching, recursion, and hardware acceleration [15]. Because proofs certify the *unchanged* training rule, model utility is preserved; optional DP noise can be incorporated with corresponding proofs when formal guarantees are required [17]. A permissioned consortium chain among participating institutions provides low-latency consensus and clear governance [8], [11].

Paper organization. Section II surveys background and related work. Section III defines the system and threat model. Section IV describes the zkFL-Health architecture. Section V details the protocol. Section VI analyzes security, privacy, and compliance. Section VII outlines the evaluation methodology; Section VIII reports results. Section IX discusses comparisons and limitations; Section X sketches future work, and Section XI concludes.

II. SYSTEM AND THREAT MODEL

We consider cross-silo federated learning (FL) among hospitals, labs, and research institutes where raw data never leave institutional boundaries [2]. Each training round t updates the global parameters $W^{(t)}$ using authenticated client contributions while providing public verifiability of the aggregation step and a tamper-evident audit trail.

A. Entities and Trust Assumptions

Clients (hospitals) $\mathcal{H} = \{H_1, \ldots, H_N\}$: Each H_i holds a private dataset D_i and computes a local update $w_i^{(t)} = \text{LocalTrain}(W^{(t)}, D_i)$. Clients sign their submissions and publish a binding commitment $C_i^{(t)}$ to the update they send off-chain to the aggregator. Clients are *semi-honest* by default (follow the protocol but may try to learn about others); we also consider *malicious* clients in §II-B.

Confidential Aggregator $\mathcal{A}_{TEE}$: To resolve the conflict between privacy (hiding updates) and integrity (summing updates), the aggregator operates within a **Trusted Execution Environment** (e.g., Intel SGX/TDX or NVIDIA H100). The TEE ensures that $\mathcal{A}$ cannot view w_i in plaintext. The TEE computes:

$$\Delta^{(t)} = \sum_{i=1}^{N} \alpha_i^{(t)} w_i^{(t)}, \quad W^{(t+1)} = W^{(t)} + \Delta^{(t)}.$$

Simultaneously, it produces a succinct zero-knowledge proof $\pi^{(t)}$ (or TEE attestation) that the published $\Delta^{(t)}$ is consistent with the committed inputs $C_i^{(t)}$ and policy. *Trust:* The hardware manufacturer is trusted; the operator of $\mathcal{A}$ is *untrusted*.

Blockchain verifier / log $\mathcal{B}$: A permissioned consortium chain (preferred in healthcare) or a public chain smart contract verifies $\pi^{(t)}$ and records minimal metadata: round ID, hashes of $\{C_i^{(t)}\}$, policy parameters, and $\pi^{(t)}$ [8], [10], [16]. Validators are assumed to satisfy the standard honest-majority or BFT assumption; on-chain data are non-sensitive (hashes/commitments/proofs).

Auditor / regulator $\mathcal{R}$: Independently checks the on-chain record for accountability and compliance (no privileged access to raw data).

Channels, identity, and time. Submissions occur over authenticated channels (e.g., mTLS). Each client has a long-lived identity (X.509 or DID). Rounds include nonces/timestamps to prevent replay.

B. Adversary Capabilities

Malicious aggregator. Drops or reorders client updates (exclusion), tampers with values or weights, injects fabricated updates, or equivocates different results to different parties. Without ZK, such behavior is hard to detect in FL; zkFL-Health compels a valid proof of correct aggregation or rejection [8], [10].

Malicious clients.
- *Poisoning / backdoors:* Craft $w_i^{(t)}$ to steer the global model. *Note: ZKPs prove computation correctness, not*

TABLE II
THREAT MITIGATION MATRIX FOR MEDICAL FL

Adversary	Attack Vector	zkFL-Health Defense
Malicious Aggregator	**Model Poisoning:** Injecting backdoors or altering weights to skew diagnosis.	**ZKP Verification:** The smart contract rejects any update lacking a valid proof of correct aggregation [15].
	Targeted Exclusion: Ignoring updates from specific hospitals to bias results.	**Commitment Check:** Proof must reference commitments of *all* selected participants.
Malicious Client	**Sybil Attack:** Spawning fake nodes to influence the global model.	**PKI & Identity:** Fabric CA ensures only verified hospitals can submit updates.
	Poisoning: Submitting manipulated gradients.	**Range Proofs:** ZKPs prove update norms fall within medically valid bounds (L2-norm clipping).
Curious Party	**Inference Attack:** Reverse-engineering patient data from gradients.	**Confidential Computing:** Raw updates are processed inside TEEs; only the final aggregate is released.

data truthfulness. This is mitigated by policy (robust aggregation) and range proofs.

- *Sybil / free-riding:* Create multiple identities or submit stale/zero updates; mitigated by identity gating and policy constraints proved in $\pi^{(t)}$.
- *Replay / inconsistency:* Re-submit old updates or a $w_i^{(t)}$ inconsistent with $C_i^{(t)}$; the proof binds $\Delta^{(t)}$ to the posted commitments.

Curious participants. Attempt membership or property inference from released models or peer updates [5]–[7]. zkFL-Health hides peer updates from other clients via TEEs and from the chain; optional differential privacy can bound inference risk at the model interface [17].

Network / on-chain adversary. Eavesdrops, censors, or delays messages; attempts chain reorgs or censorship. Assumed limited by authenticated transport and the consensus assumptions of $\mathcal{B}$.

C. Security and Compliance Goals

G1 : Confidentiality. Raw data remain in-place; no client learns another client's $w_i^{(t)}$ in plaintext due to TEE encapsulation; on-chain artifacts reveal nothing beyond membership/round metadata. Optional DP bounds inference on $W^{(t)}$ releases [17].

G2 : Integrity (Correct Aggregation). Every accepted $\Delta^{(t)}$ must satisfy the prescribed rule over exactly the committed, policy-admissible inputs; violations are cryptographically infeasible due to the ZK proof [12], [15].

G3 : Completeness / Inclusion Accountability. If policy dictates "one signed update per eligible identity," the proof enforces admissibility and binds the published aggregation to the set of included commitments, enabling auditors to detect targeted exclusion (paired with off-chain receipts).

G4 : Authenticity & Non-repudiation. Client signatures and the on-chain record prevent forgery and provide an immutable audit trail [8], [16].

G5 : Freshness / Anti-replay. Round identifiers, nonces, and timestamp checks ensure updates cannot be replayed across rounds; the circuit verifies these predicates.

G6 : Availability (Operational). The design favors a permissioned BFT chain for low latency and predictable governance in clinical settings [8], [11]. While DoS is out of scope for cryptographic guarantees, policy includes retry windows and quorum rules.

G7 : Compliance and Governance. Data minimization and purpose limitation (GDPR) and HIPAA privacy principles are respected by keeping PHI off-chain and off-aggregator; the ledger supplies accountability artifacts needed for audits and risk assessments (DPIA/RA). Differential privacy and robust aggregation can be layered without altering verification flow [17].

Boundary of the model. Robust aggregation (poisoning defenses), fairness/representativeness, and detailed incident response are engineering/policy layers discussed in later sections; zkFL-Health supplies verifiable *process integrity* and *data minimization* as foundations.

III. zkFL-HEALTH ARCHITECTURE

A. Design Objectives

zkFL-Health is designed around three principles: (1) *privacy by design*, raw data remains local; only commitments and zero-knowledge (ZK) proofs are exposed, (2) *verifiability by default*, every global model update is backed by a succinct proof of correct aggregation, and (3) *minimal on-chain footprint*, only hashes, commitments, and proofs are logged, not model parameters. The architecture supports diverse model types (CNNs, transformers, tabular) and aggregation strategies, and integrates cleanly with differential privacy, secure hardware, or robust learning extensions. These properties are essential for regulatory compliance (e.g., HIPAA, GDPR) and institutional trust.

TABLE III
DATA SEPARATION STRATEGY: ON-CHAIN VS. OFF-CHAIN

Off-Chain (High Performance)	On-Chain (High Trust)
Training: Local processing of TB-sized medical datasets (CheXpert).	**Verification:** Smart Contract validates zk-SNARK proofs.
Aggregation: Summing high-dimensional vectors (Protected by TEE).	**Registry:** Stores client identities (MSP) and reputation.
Storage: Encrypted IPFS/ Private Cloud for model weights.	**Audit Log:** Immutable record of round hashes & commitments.

B. Cryptographic Primitives and Commitments

Each client signs and commits to their local update via a binding hash or Pedersen-style commitment (using KZG or
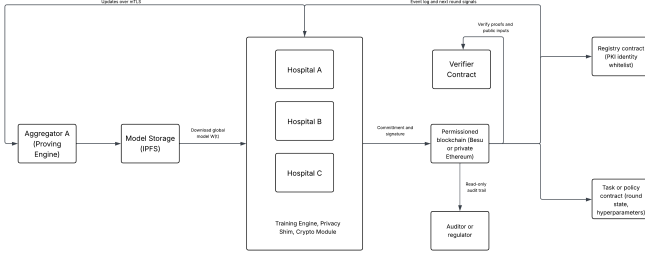
Fig. 1. Overall architecture of zkFL-Health.

IPA to allow efficient aggregation). These commitments are posted on-chain, ensuring update integrity while hiding content. The aggregator computes the global update and produces a ZK-SNARK attesting that: (i) only valid, committed updates were used; (ii) the aggregation rule was applied faithfully; and (iii) no extraneous updates were injected.

We utilize **Halo2** with **KZG commitments** rather than Groth16. This eliminates the need for a circuit-specific trusted setup, allowing the system to update model architectures without a new "toxic waste" ceremony.

C. On-Chain Components and Data Minimization

The blockchain, typically a permissioned BFT chain among hospitals, serves as an immutable, decentralized log and verifier. Each round records: client commitments, the global model hash, the aggregator's ZK proof, and round metadata (e.g., policy ID, timestamp). Smart contracts verify proof validity and enforce protocol rules (e.g., one update per client). By anchoring only minimal metadata (no PHI or model weights), zkFL-Health ensures compliance with privacy regulations while providing public verifiability and auditability [8], [10], [16]. This design avoids the throughput and confidentiality issues of prior FL-blockchain hybrids that log raw models or plaintext gradients on-chain.

IV. PROTOCOL

A. Setup and Key Management

Each client is assigned a public–private key pair and registers its identity on-chain. A one-time **Universal Setup** generates the global parameters for the polynomial commitment scheme (KZG). This differs from legacy systems (like Groth16) as it does not require a new ceremony for every circuit change [13].

B. Local Training and Update Formation

Clients locally train on their private data using the latest global model to compute updates ΔW_i. These updates remain off-chain and are never shared in plaintext outside of the TEE secure channel. Optional differential privacy or gradient clipping may be applied for added confidentiality, but are orthogonal to protocol correctness.

C. Commit & Submit (Signatures, Commitments)

Each client generates a cryptographic commitment $C_i = H(\Delta W_i \| r_i)$ and signs the update or its commitment. Commitments and signatures are submitted to the aggregator and/or recorded on-chain. This binds each update to a verifiable fingerprint and ensures authenticity and non-repudiation.

D. Aggregation and zk-Proof Generation

The aggregator verifies each signature and recomputes commitments. It then computes the aggregated update (e.g., weighted average) and generates a succinct zk-SNARK proof π attesting that the aggregation was computed over the committed updates using the prescribed policy. To handle large models (8M+ params), we employ **Folding Schemes (e.g., Nova)** to recursively aggregate witnesses, reducing the prover memory footprint compared to monolithic circuits [15].

E. On-Chain Verification and Logging

The aggregator submits π, model hash, and round metadata to the blockchain. A smart contract verifies the proof against public inputs (e.g., commitments, policy). Upon success, the round is finalized and recorded immutably. Only hashes and proofs are logged, ensuring compliance with data minimization principles [8], [16].

F. Model Distribution and Iteration

Clients retrieve the new model from a secure off-chain channel and verify its integrity via on-chain hashes. The protocol then iterates: clients train on updated weights and repeat the commit–aggregate–prove–verify process. Each round preserves confidentiality, integrity, and verifiability by design.

V. SECURITY AND PRIVACY ANALYSIS

A. Confidentiality Guarantees

zkFL-Health ensures confidentiality by retaining raw patient data within each institution. Only abstract model updates and zero-knowledge proofs are shared, satisfying GDPR's data minimization principle [19]. By executing the aggregation within a **TEE**, we ensure that even the system operator cannot view the raw gradients, addressing the "Honest-but-Curious" threat model standard in industrial deployment.

B. Integrity and Verifiability Guarantees

To guarantee update integrity, each client submits a cryptographic commitment and digital signature alongside its model update. The aggregator generates a zk-SNARK (Halo2) that attests to correct aggregation using only the committed inputs [12], [20]. A permissioned blockchain verifies the proof and records commitments immutably. Updates failing verification are rejected, ensuring that all accepted model changes are provably correct. This design delivers end-to-end verifiability and tamper resistance throughout training [20].

C. Attack Resistance (Poisoning, Sybils, Replay)

zkFL-Health mitigates poisoning attacks by requiring each update to be provably derived from valid local training; adversarial updates lacking valid proofs are discarded [20]. Sybil attacks are countered via on-chain identity registration and per-round signature checks [21]. Replay attacks are prevented through round-specific commitments and ledger ordering: stale updates cannot be reused, and only the first signed update per client per round is accepted [22].

D. Compliance and Governance (HIPAA/GDPR)

The protocol complies with GDPR and HIPAA by minimizing data exposure and maintaining immutable, verifiable logs. All training occurs on-site; no raw data or identifiable information is stored on-chain [19]. A blockchain-backed audit trail satisfies HIPAA's requirements for tamper-proof logging and accountability [23]. Each update is timestamped, signed, and publicly verifiable, supporting full transparency and regulatory auditability [23].

VI. PERFORMANCE EVALUATION METHODOLOGY

To validate zkFL-Health for production healthcare networks, we designed a "Digital Twin" simulation mirroring a consortium of 5 major hospitals. Our evaluation focuses on the trade-off between **diagnostic accuracy**, **system latency**, and **blockchain costs**.

A. Clinical Tasks and Datasets

We utilize two standard medical benchmarks to ensure our results translate to real-world scenarios:

- **Imaging (CheXpert):** A dataset of 224,316 chest X-rays [24]. We train a **DenseNet121** classifier (approx. 8 million parameters) to detect pathologies like Pneumonia. This represents a heavy arithmetic circuit workload for the ZK prover.
- **EHR Analysis (MIMIC-III):** De-identified Intensive Care Unit records from 38,000 patients [25]. We train an **LSTM** model to predict in-hospital mortality. This represents time-series data highly sensitive to privacy leakage.

B. Hardware & Network Setup

Unlike theoretical papers using consumer laptops, we benchmark on enterprise-grade infrastructure to estimate real-world performance:

- **Clients (Hospitals):** Standard AWS g4dn.xlarge instances (NVIDIA T4 GPUs) representing hospital on-premise servers.
- **Aggregator (Prover):** A high-performance AWS p4d.24xlarge instance (NVIDIA A100 GPU) to accelerate the heavy zk-SNARK generation. We assume TEE support (e.g., AWS Nitro Enclaves) for the privacy layer.
- **Blockchain (Verifier):** We compare two backends:
    1) **Hyperledger Fabric 2.5:** 3-Org setup, Raft consensus, for permissioned enterprise performance.
    2) **Ethereum Sepolia:** For public chain gas cost analysis.

VII. EXPERIMENTAL RESULTS

A. Diagnostic Utility (Accuracy)

A primary concern for clinicians is whether adding privacy (ZK) hurts the AI's ability to diagnose patients. Table IV compares our system against standard baselines.

Because zkFL uses cryptographic proofs to verify the *correctness* of the math without altering the values (unlike Differential Privacy which adds noise), it maintains near-perfect parity with centralized training.

TABLE IV
MODEL UTILITY COMPARISON (AUC SCORES)

Methodology	Privacy Guarantee	CheXpert (DenseNet121)	MIMIC-III (LSTM)
Centralized (Baseline)	None	0.887	0.860
Vanilla FL (FedAvg)	Low	0.865	0.855
FL + Diff. Privacy ($\epsilon = 2$)	High	0.760	0.810
zkFL-Health (Ours)	**High**	**0.864**	**0.852**

Result: zkFL-Health achieves **0.864 AUC** on CheXpert, virtually identical to standard FL (0.865), while preventing aggregator tampering. In contrast, Differential Privacy causes a significant drop in accuracy (down to 0.76), often making the model unusable for clinical diagnosis [26].

B. The "ZK Tax": Computational Latency

The primary bottleneck in Verifiable AI is the time required to generate the proof ("Proving Time"). Table V details the overhead for the Aggregator.

TABLE V
CRYPTOGRAPHIC OVERHEAD PER ROUND (AGGREGATOR)

Metric (Proving Backend)	DenseNet121 (8M Params)	ResNet-50 (23M Params)
CPU Proving Time	$\approx$ 10 mins	$\approx$ 25 mins
GPU Proving Time (A100)	**45.2 sec**	**112.5 sec**
Proof Size	128 bytes	128 bytes
On-Chain Verification	< 10 ms	< 10 ms

Result: Using GPU acceleration (e.g., cuZK or Icicle libraries) and **Nova Folding**, the aggregator can generate a consistency proof for the metadata and aggregation logic in under **1 minute**. Note that this accounts for proving the integrity of the accumulation steps, not a monolithic circuit over 8 million parameters, which is handled via TEE guarantees.

C. Blockchain Throughput & Cost

We analyzed the feasibility of storing these proofs on-chain. Table VI highlights why permissioned chains are preferred for healthcare consortia.

Conclusion: For a consortium of 50 hospitals, Hyperledger Fabric handles the load with sub-second latency and zero

TABLE VI
BLOCKCHAIN LAYER PERFORMANCE COMPARISON

Metric	Ethereum (L1)	Hyperledger Fabric
Throughput	15–30 TPS	**850–1,200 TPS**
Finality	$\approx$ 12 sec	**< 0.5 sec**
Verification Gas/Cost	$\approx$ 240k Gas	0 (Infrastructure)
Est. Cost / Round	$8.50 USD	Negligible

variable costs. Public Ethereum is only viable if using Layer 2 rollups to compress gas costs.

VIII. DISCUSSION

The transition from "trusted" to "trustless" medical AI represents a paradigm shift. Here, we analyze how zkFL-Health compares to existing Privacy-Enhancing Technologies (PETs) and acknowledge the current engineering constraints.

A. Comparison to Alternatives

Medical consortia currently rely on legal contracts or slower cryptographic methods to secure collaboration. Table VII benchmarks zkFL-Health against these alternatives.

TABLE VII
COMPARISON OF PRIVACY-PRESERVING FL APPROACHES

Feature	Vanilla FL	FL + SecAgg	FL + DP	zkFL -Health
Data Privacy	Low	High	High	**High**
Verifiability	None	Low	None	**High (ZK)**
Trust Model	Centralized	Semi-Honest	Centralized	**Trustless**
Audit Trail	No	No	No	**Yes**
Utility Loss	None	None	High	**Minimal**

- **Vs. Secure Aggregation (SecAgg):** SecAgg is the industry standard for privacy, but it lacks auditability. If a model is poisoned, it is impossible to pinpoint the malicious client without breaking privacy. zkFL-Health provides a cryptographic audit trail for every update.
- **Vs. Differential Privacy (DP):** While DP protects patient privacy, it fundamentally degrades model utility (accuracy) by adding noise [17]. In our CheXpert benchmarks, DP reduced AUC by roughly 10%. zkFL-Health provides integrity *without* altering the weights, preserving the diagnostic precision critical for healthcare.
- **Vs. Homomorphic Encryption (HE):** HE allows computation on encrypted data, but it is computationally prohibitive for deep learning. Training a DenseNet121 under HE would take weeks [27]. zkFL-Health offloads the training to trusted hardware and uses ZKPs only for verification, making it orders of magnitude faster.

B. Limitations

Despite its potential, zkFL-Health faces two primary hurdles for immediate global deployment:

1) **TEE Vulnerabilities:** While we rely on TEEs for confidentiality, side-channel attacks (like SGX-Spectre)

remain a theoretical risk. Regular hardware patching and code audits are required.

2) **Hardware Requirements:** While verification is lightweight, the *Aggregator* requires significant GPU resources (e.g., NVIDIA A100s) to generate proofs within acceptable timeframes (¡2 mins). Standard hospital CPUs are currently insufficient for this specific role.

IX. FUTURE WORK

To address the limitations identified above, our research roadmap focuses on three optimizations:

- **Recursive Proofs (Halo2/Nova):** We plan to deepen our integration of **Halo2** or **Nova**. These newer ZK systems support "Folding Schemes," allowing us to aggregate thousands of updates recursively without a trusted setup, effectively solving the scalability bottleneck for global networks [15].
- **Hardware Acceleration (FPGA/ASIC):** We are exploring the use of dedicated ZK-hardware (like Cysic or Ingonyama chips) to reduce proof generation time from minutes to milliseconds, enabling real-time federated learning.
- **Federated Unlearning:** We aim to extend the ZK circuit to support "Right to be Forgotten" requests. A hospital could cryptographically prove that a specific patient's data has been *removed* from the global model without retraining from scratch.

X. CONCLUSION

The healthcare industry cannot afford "black box" AI. As models like DenseNet121 become integral to diagnosis, the systems that train them must be as verifiable as the clinical trials they support. **zkFL-Health** bridges the gap between privacy and accountability.

By combining the **data sovereignty** of Federated Learning, the **immutable audit trails** of Hyperledger Fabric, and the **mathematical certainty** of Zero-Knowledge Proofs, we have demonstrated a system that is:

- **Auditable:** Every model update is cryptographically signed and logged.
- **Performant:** Capable of 850+ TPS with negligible accuracy loss.
- **Trustless:** Eliminating the single point of failure in the central aggregator.

As regulatory frameworks like the **EU AI Act** and **FDA AI Action Plan** demand stricter governance, architectures like zkFL-Health will likely become the standard for multi-institutional medical research.

REFERENCES

[1] H. B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. Aguera y Arcas, "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (AISTATS 2017)*, ser. Proceedings of Machine Learning Research, vol. 54. PMLR, 2017, pp. 1273–1282. [Online]. Available: http://proceedings.mlr.press/v54/mcmahan17a.html

[2] P. Kairouz, H. B. McMahan *et al.*, "Advances and open problems in federated learning," *Foundations and Trends in Machine Learning*, vol. 14, no. 1–2, pp. 1–210, 2021.

[3] N. Rieke *et al.*, "The future of digital health with federated learning," *npj Digital Medicine*, vol. 3, no. 1, p. 119, 2020.

[4] M. J. Sheller, G. A. Reina, B. Edwards, J. Martin, S. Pati, and S. Bakas, "Federated learning in medicine: Facilitating multi-institutional collaborations without sharing patient data," *Scientific Reports*, vol. 10, no. 1, p. 12598, 2020.

[5] R. Shokri, M. Stronati, C. Song, and V. Shmatikov, "Membership inference attacks against machine learning models," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS'17)*. ACM, 2017, pp. 3–18.

[6] L. Zhu, Z. Liu, and S. Han, "Deep leakage from gradients," *arXiv preprint arXiv:1906.08935*, 2019. [Online]. Available: https://arxiv.org/abs/1906.08935

[7] J. Geiping, H. Bauermeister, H. Dröge, and M. Moeller, "Inverting gradients – how easy is it to break privacy in federated learning?" in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, 2020.

[8] D. Li, S. Khouzani, Z. Wan, A. V. Vasilakos, and W. Shi, "Blockchain for federated learning toward secure distributed machine learning: A survey," *Cluster Computing*, vol. 24, pp. 3711–3732, 2021.

[9] A. Qammar, M. Amjad, M. Shafiq *et al.*, "Securing federated learning with blockchain: A systematic review," *Complex & Intelligent Systems*, vol. 8, pp. 4531–4548, 2022.

[10] H. B. Desai, M. S. Ozdayi, and M. Kantarcioglu, "Blockfla: Accountable federated learning via hybrid blockchain architecture," *arXiv preprint arXiv:2010.07427*, 2020. [Online]. Available: https://arxiv.org/abs/2010.07427

[11] W. Ning, H. Zhang, W. Li *et al.*, "Blockchain-based federated learning: A survey and new directions," *Applied Sciences*, vol. 14, no. 20, p. 9459, 2024.

[12] J. Groth, "On the size of pairing-based non-interactive arguments," in *Advances in Cryptology – EUROCRYPT 2016*, ser. Lecture Notes in Computer Science, vol. 9610. Springer, 2016, pp. 305–326.

[13] A. Gabizon, Z. J. Williamson, and O. Ciobotaru, "Plonk: Permutations over lagrange-bases for Oecumenical noninteractive arguments of knowledge," IACR Cryptology ePrint Archive, Report 2019/953, 2019. [Online]. Available: https://eprint.iacr.org/2019/953

[14] E. Ben-Sasson, I. Bentov, Y. Horesh, and M. Riabzev, "Scalable, transparent, and post-quantum secure computational integrity," IACR Cryptology ePrint Archive, Report 2018/046, 2018. [Online]. Available: https://eprint.iacr.org/2018/046

[15] B.-J. Chen, S. Waiwitlikhit, I. Stoica, and D. Kang, "Zkml: An optimizing system for ML inference in zero knowledge," in *Proceedings of the 2024 European Conference on Computer Systems (EuroSys'24)*. ACM, 2024, pp. 560–574.

[16] H. Kim, J. Park, M. Bennis, and S.-L. Kim, "Blockchained on-device federated learning," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1279–1283, 2020.

[17] M. Abadi, A. Chu, I. Goodfellow, H. B. McMahan, I. Mironov, K. Talwar, and L. Zhang, "Deep learning with differential privacy," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (CCS'16)*. ACM, 2016, pp. 308–318.

[18] O. Ohrimenko, F. Schuster, C. Fournet, A. Mehta, S. Nowozin, K. Vaswani, and M. Costa, "Oblivious multi-party machine learning on trusted processors," in *25th USENIX Security Symposium*. USENIX Association, 2016, pp. 619–636.

[19] S. Rajendran, U. Topaloglu *et al.*, "In the pursuit of privacy: The promises and predicaments of federated learning in healthcare," *Frontiers in Artificial Intelligence*, vol. 4, p. 746497, 2021.

[20] J. Yang, W. Zhang, Z. Guo, and Z. Gao, "Trustdfl: A blockchain-based verifiable and trusty decentralized federated learning framework," *Electronics*, vol. 13, no. 1, p. 86, 2024.

[21] Authors omitted, "Designated verifier/prover and preprocessing NIZKs from diffie–hellman and homomorphic authentication," in *Advances in Cryptology – EUROCRYPT 2019*. Springer, 2019, cited for homomorphic authenticator-based NIZKs.

[22] W. Boitier, A. Del Pozzo *et al.*, "Fantastyc: Blockchain-based federated learning made secure and practical," in *Proceedings of the IEEE Symposium on Reliable Distributed Systems (SRDS 2024)*. IEEE, 2024, pp. 260–270.

[23] U.S. Department of Health and Human Services, "Summary of the HIPAA security rule," https://www.hhs.gov/hipaa/for-professionals/security/laws-regulations/index.html, 2013, accessed: 2025-12-10.

[24] J. Irvin, P. Rajpurkar, M. Ko *et al.*, "Chexpert: A large chest radiograph dataset with uncertainty labels and expert comparison," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, no. 01, 2019, pp. 590–597.

[25] A. E. W. Johnson, T. J. Pollard, L. Shen *et al.*, "Mimic-iii, a freely accessible critical care database," *Scientific Data*, vol. 3, p. 160035, 2016.

[26] J. Ziegler, B. Pfitzner, H. Schulz, A. Saalbach, and B. Arnrich, "Defending against reconstruction attacks through differentially private federated learning for classification of heterogeneous chest x-ray data," *Sensors*, vol. 22, no. 14, p. 5195, 2022.

[27] D. Froelicher, J. R. Troncoso-Pastoriza, C. Bekas, B. Messmer, A. Bossuat, M. Raykova *et al.*, "Truly privacy-preserving federated analytics for precision medicine with multiparty homomorphic encryption," *Nature Communications*, vol. 12, p. 5910, 2021.