Blockchain-Integrated Trust Management for Backdoor-Resistant Distributed Large Language Model Ecosystems

Cesar Gregory
School of Electrical Engineering and Computer Science, Oregon State University, Corvallis, OR, USA.
contactcesar@oregonstate.edu

Nikhil Jha
Department of Computer Science, Binghamton University, Binghamton, NY, USA.
hellonikhil@binghamton.edu

Abstract

The rapid proliferation of large language models across decentralized infrastructures has introduced unprecedented challenges in ensuring model integrity, trustworthiness, and resilience against adversarial manipulations. Backdoor attacks, wherein malicious actors embed hidden triggers that cause targeted misbehavior, pose a particularly insidious threat in distributed ecosystems where multiple stakeholders contribute data, compute, or model updates. This paper proposes a blockchain-integrated trust management framework designed specifically for backdoor-resistant distributed large language model ecosystems. The framework leverages immutable ledger properties, smart contract-based governance, and decentralized identity to establish verifiable provenance for training data, model checkpoints, and inference outputs. We examine architectural trade-offs between transparency, latency, and scalability, and discuss how consensus mechanisms can be adapted to support heterogeneous participants with varying trust levels. A hybrid on-chain and off-chain verification strategy is introduced to reconcile the computational overhead of blockchain operations with the high-throughput requirements of large language model deployment. The paper further addresses backdoor resistance through a combination of cryptographic attestation, differential privacy, and prototype-based anomaly detection, drawing on recent advances in vertical split learning defense. Governance mechanisms such as staking, slashing, and reputation scoring are analyzed in the context of aligning incentives for honest behavior. Deployment considerations including energy consumption, regulatory compliance, and cross-chain interoperability are critically assessed. By synthesizing insights from distributed systems, cryptoeconomics, and machine learning security, this work provides a comprehensive blueprint for building trustworthy and resilient large language model ecosystems that are resistant to backdoor attacks while maintaining operational efficiency and fairness.

Keywords

blockchain, trust management, backdoor resistance, large language model, distributed ecosystem, smart contract, decentralized governance, provenance, differential privacy, vertical split learning.

1. Introduction

The emergence of large language models as foundational components of modern artificial intelligence systems has shifted the paradigm from centralized model training and deployment to increasingly distributed and collaborative arrangements. Organizations and individuals now participate in federated learning, split learning, and model marketplace ecosystems where data, computation, and model parameters are shared across loosely coupled networks. While such decentralization offers benefits in terms of data privacy, resource pooling, and reduced single points of failure, it also broadens the attack surface for malicious actors seeking to compromise model behavior through backdoor attacks. A backdoor attack involves embedding a hidden pattern in the training data or model updates such that the model behaves normally on benign inputs but produces attacker-chosen outputs when a specific trigger is present. The distributed nature of these ecosystems makes it difficult to attribute malicious contributions, verify the integrity of model components, and establish a unified trust baseline across heterogeneous participants.

Traditional trust management approaches, such as centralized certificate authorities or reputation systems, are often ill-suited for dynamic, permissionless, or semi-trusted environments where participants may join and leave at will. Blockchain technology, with its properties of immutability, transparency, and decentralized consensus, offers a promising foundation for building trust in distributed machine learning pipelines. By recording critical metadata, model hashes, and access control policies on a distributed ledger, blockchain can provide a tamper-evident audit trail that enables retrospective verification and accountability. However, the integration of blockchain with large language model ecosystems is not straightforward due to the high computational and storage costs associated with on-chain operations, as well as the latency constraints of real-time inference.

This paper presents a systematic exploration of blockchain-integrated trust management tailored for backdoor-resistant distributed large language model ecosystems. We propose a layered architecture that separates the verification of model integrity from the execution of training and inference, allowing efficiency-critical operations to occur off chain while maintaining an immutable record of commitments and proofs. We further examine how smart contracts can automate trust policies, such as requiring proof of data provenance before accepting contributions, and how incentive mechanisms can deter malicious behavior. A key contribution is the incorporation of recent advances in backdoor defense, specifically prototype-based anomaly detection for vertical split learning, into the trust framework. This approach enables distributed participants to collectively verify the consistency of learned representations without revealing private data, thereby preserving privacy while enhancing security.

The paper proceeds as follows. Section 2 reviews related work in trust management, blockchain for machine learning, and backdoor attacks. Section 3 describes the proposed system architecture and trust framework. Section 4 details the blockchain integration for verifiable provenance. Section 5 discusses backdoor resistance mechanisms. Section 6 addresses governance and incentive design. Section 7 explores deployment considerations, scalability trade-offs, and fairness. Section 8 concludes with future directions.

2. Background and Related Work

Trust management in distributed systems has been extensively studied, with early approaches relying on reputation systems, web of trust, and centralized identity providers. More recently, blockchain-based identity and access management solutions have been proposed to decentralize trust without requiring a single authority [1]. In the context of machine learning,

federated learning frameworks often employ secure aggregation and differential privacy to protect participant privacy, but these methods do not inherently address model integrity or backdoor resistance [2]. Several works have explored using blockchain to record model updates and verify contributions in federated learning settings [3,4]. These systems typically use smart contracts to manage reward distribution and detect anomalous updates based on statistical metrics. However, most existing frameworks focus on horizontal federated learning where participants share the same feature space, and they do not consider the unique vulnerabilities of large language models, such as embedding-layer attacks or distributed backdoors across heterogeneous data splits.

Backdoor attacks have been studied extensively in both centralized and distributed settings. In centralized training, triggers can be embedded by corrupting a small fraction of the training data [5]. In distributed settings, malicious participants can submit poisoned model updates that are aggregated with honest updates, making detection more challenging [6]. Defenses include pruning, fine-tuning, and differential privacy-based sanitization. For vertical split learning, where parties hold different feature subsets and collaborate through a split model architecture, backdoor attacks can be particularly stealthy because the adversary controls only part of the model and can influence intermediate representations [7]. Recent work by Shui et al. introduced ProtoGuard-SL, a prototype consistency based backdoor defense for vertical split learning, which leverages the idea that the learned prototypes of honest participants should remain consistent even when an adversary injects a trigger [8]. This defense operates by comparing prototype vectors across different splits and flagging inconsistencies, without requiring access to raw data. Our trust framework incorporates such prototype verification as an off-chain attestation that can be anchored on the blockchain.

Blockchain scalability remains a challenge for high-throughput applications. Layer-2 solutions, sharding, and sidechains have been proposed to increase transaction throughput while maintaining security [9]. For machine learning applications, storing entire model weights on chain is infeasible; instead, cryptographic hashes and zero-knowledge proofs are used to verify that a model update corresponds to an approved computation [10]. Our architecture adopts a hybrid approach where only commitment proofs and attestation summaries are stored on chain, while bulk data remains off chain in decentralized storage networks such as IPFS or in encrypted databases.

3. System Architecture and Trust Framework

The proposed system is organized around a set of roles: model owners, data contributors, compute providers, validators, and end users. Each participant is identified by a decentralized identifier (DID) registered on the blockchain, which is bound to a public key and a set of attributes attested by credential issuers. Trust is established through a combination of identity verification, behavioral reputation, and cryptographic proofs. The system operates in epochs, where each epoch corresponds to a training round, a batch of inferences, or a model update cycle. At the beginning of each epoch, smart contracts define the terms of participation, including data quality requirements, computational resource commitments, and expected behavior. Participants submit commitments (e.g., hashes of their data contributions or model updates) to the blockchain, which are timestamped and made public.

During the epoch, actual training or inference occurs off chain using secure multi-party computation or trusted execution environments, depending on the sensitivity of the data. Validators periodically request attestation proofs from participants. These proofs can be in the form of zero-knowledge proofs of correct computation, or in the case of vertical split learning,

prototype consistency checks. Once validated, the results are aggregated and the final model state or inference output is recorded as a hash on the blockchain. Any dispute or suspected backdoor incident triggers an audit, during which the blockchain provides an immutable log of all commitments and attestations for forensic analysis.

The architecture employs a two-tier trust model. Tier one consists of high-trust participants, such as established institutions that undergo regular audits and have a long positive reputation. Tier two includes new or low-stake participants who must provide collateral in the form of staked tokens and accept higher scrutiny. This tiered approach balances the need for openness with the requirement for robustness against Sybil attacks. The blockchain consensus mechanism is chosen based on the expected scale and latency requirements: for permissioned consortiums, Byzantine fault-tolerant protocols like Tendermint are suitable; for permissionless settings, proof-of-stake with built-in slashing conditions is preferred.

4. Blockchain Integration for Verifiable Provenance

Verifiable provenance is critical for backdoor resistance because it allows any participant to trace the origin of a data sample, a model weight, or an inference output. In our framework, provenance data is recorded in the form of directed acyclic graphs (DAGs) that link inputs, transformations, and outputs. Each node in the graph is associated with a cryptographic hash, and edges are signed by the participant responsible for the transformation. The root hash of the DAG is periodically committed to the blockchain as a Merkle tree root, enabling efficient verification of any subgraph without revealing the entire provenance chain.

Smart contracts manage access control and permissioned verification. For instance, a data contributor can grant access to their provenance graph only to authorized validators via encrypted pointers stored on chain. The blockchain also records metadata such as the timestamp, the identity of the participant, and the type of operation (e.g., data preprocessing, gradient update, inference). This metadata can be used to detect statistical anomalies, such as an unusually high number of updates from a single participant within a short time window, which may indicate a coordinated backdoor injection attempt.

One challenge is the storage overhead of provenance graphs. To address this, we employ a hierarchical structure: high-level summaries (e.g., aggregate statistics) are stored on chain, while detailed provenance data resides on off-chain decentralized storage. The blockchain contains pointers and hashes that ensure the integrity of the off-chain data. Additionally, periodic checkpoints are created where the full provenance graph is hashed and stored in batch. This reduces on-chain transaction frequency while preserving auditability.

5. Backdoor Resistance Mechanisms

The trust framework incorporates multiple layers of defense against backdoor attacks. First, at the data contribution stage, participants are required to submit a zero-knowledge proof that their data satisfies certain properties, such as not containing specific trigger patterns, without revealing the actual data. This is computationally intensive but can be amortized over multiple contributions. Second, during model training, differential privacy mechanisms are applied to gradient updates, limiting the influence any single participant can have on the model. However, differential privacy alone may not remove a well-crafted backdoor because the trigger can be designed to affect only a small fraction of outputs; therefore, additional verification is necessary.

Third, for distributed architectures that involve vertical split learning, we adopt a prototype consistency verification method [8]. After each training round, each party computes prototype vectors that represent the centroids of learned feature representations for each class. These prototypes are encrypted and shared with validators, who compare the consistency between different parties. Significant divergence indicates a potential backdoor injection, as the malicious party's prototype will deviate from the honest prototype. This verification is efficient because it operates on aggregated statistics rather than individual data points. The resulting attestation proofs are recorded on the blockchain as evidence of model integrity.

Fourth, an on-chain anomaly detection smart contract monitors the frequency and pattern of prototype consistency violations. If the same participant is flagged repeatedly, the smart contract automatically triggers a penalty, such as slashing part of their staked tokens or revoking their participation privileges. This creates a strong deterrent against attempting to inject backdoors. The detection threshold is dynamically adjusted based on the overall network condition and the reputation of the participant, following a game-theoretic analysis that balances false positive rates and security levels.

6. Governance and Incentive Design

Governance in a distributed large language model ecosystem must address the tension between openness and security. Our framework employs a decentralized autonomous organization (DAO) structure where token holders vote on key parameters such as the required staking amounts, the frequency of audits, and the criteria for validator selection. Smart contracts execute these policies automatically. To incentivize honest behavior, participants earn reputation tokens for each successful validation and contribution. Reputation tokens confer voting power and access to higher-tier rewards. Conversely, malicious behavior results in token slashing and reputation penalties.

Incentive alignment also requires that validators are compensated for their verification work. Fees are collected from model owners and distributed among validators based on their performance and consistency. To prevent collusion between validators and malicious participants, a rotating committee of validators is selected randomly each epoch, and their identities are revealed only after the epoch ends. This commit-reveal scheme reduces the risk of bribery. Furthermore, any validator who fails to detect a backdoor that is later discovered can be held accountable through a dispute resolution mechanism that relies on economic penalties.

The governance system must also consider fairness across participants with different resource endowments. Small contributors may not be able to provide large stakes, yet their participation is valuable for diversity. A tiered contribution system allows them to join with smaller stakes and correspondingly lower rewards, but with a higher frequency of audits. This ensures that even low-resourced participants can contribute without introducing disproportionate risk.

7. Deployment Considerations and Trade-offs

Deploying a blockchain-integrated trust management system for large language model ecosystems involves several practical trade-offs. The most significant is the latency overhead introduced by on-chain transactions. Each epoch requires multiple commits and verification steps, which can take seconds to minutes depending on the blockchain's block time. For real-time inference applications, this latency is unacceptable. Therefore, our framework distinguishes between critical trust events (such as model updates) and routine inference

requests. Only model update epochs are subject to full blockchain verification; inference results are verified via lighter-weight checks that leverage the already verified model state.

Another trade-off is between transparency and privacy. While the blockchain provides an open ledger that all participants can inspect, revealing too much information about model parameters or data can enable adversarial reverse engineering. We balance these by using zero-knowledge proofs and homomorphic commitments that allow verification without disclosure. Furthermore, participants can choose to operate under a permissioned blockchain that restricts read access to authorized parties. In regulated industries such as healthcare or finance, compliance with data protection laws like GDPR may require the ability to delete personal data, which conflicts with blockchain immutability. To address this, we implement off-chain storage with cryptographic hashes on chain, allowing data to be removed from off-chain repositories while preserving the integrity proof. The hash remains on chain as evidence that the data existed at a certain time, but without exposing the content.

Energy consumption is another concern, especially for proof-of-work based blockchains. Our framework is designed to operate on proof-of-stake or delegated proof-of-stake consensus mechanisms, which have a much lower carbon footprint. Additionally, validators can be incentivized to use renewable energy sources through green token rewards. Cross-chain interoperability is essential for ecosystems that span multiple blockchains; we employ relay chains and atomic swaps to synchronize trust state across different ledgers.

Finally, we must consider the scalability of the prototype consistency verification method [8] as the number of parties grows. The pairwise comparison of prototypes scales quadratically, but can be approximated through clustering and sampling. In large ecosystems, a hierarchical verification tree can be used where local groups verify one another, and group leaders verify across groups. This reduces the overall verification complexity while maintaining robustness.

8. Conclusion

This paper has presented a comprehensive framework for blockchain-integrated trust management designed to render distributed large language model ecosystems resistant to backdoor attacks. By combining immutable provenance recording, smart contract-based governance, and advanced backdoor detection techniques such as prototype consistency verification, the proposed architecture addresses the unique security challenges that arise when multiple heterogeneous participants collaborate on training and deploying large models. The hybrid on-chain/off-chain design reconciles the need for transparency and accountability with the performance demands of large-scale machine learning. We have also discussed the critical trade-offs between latency, privacy, scalability, and energy efficiency, and suggested design choices that can be tailored to different deployment contexts.

Future research directions include the empirical evaluation of the framework on real-world distributed training platforms, the development of formal game-theoretic models for incentive design, and the exploration of adaptive consensus mechanisms that dynamically adjust security parameters based on detected threat levels. Additionally, integrating zero-knowledge machine learning techniques to further reduce validation overhead remains an open problem. As large language models become ever more embedded in critical infrastructures, the need for robust, decentralized trust management will only intensify, and the principles laid out in this paper provide a foundation for building such systems.

References

1. Nakamoto, S. (2008). Bitcoin: A peer-to-peer electronic cash system.

2. Bonawitz, K., Eichner, H., Grieskamp, W., Huba, D., Ingerman, A., Ivanov, V., ... & Zhang, C. (2019). Towards federated learning at scale: System design. Proceedings of Machine Learning and Systems, 1, 374-388.

3. Li, Y., Zhou, Y., & Chen, C. (2022). Blockchain-based federated learning with decentralized trust. IEEE Transactions on Network and Service Management, 19(4), 4526-4540.

4. Weng, J., Weng, J., Zhang, J., Li, M., Zhang, Y., & Luo, W. (2021). DeepChain: Auditable and privacy-preserving deep learning with blockchain-based incentive. IEEE Transactions on Dependable and Secure Computing, 18(5), 2046-2060.

5. Gu, T., Dolan-Gavitt, B., & Garg, S. (2017). BadNets: Identifying vulnerabilities in the machine learning model supply chain. arXiv preprint arXiv:1708.06733.

6. Bagdasaryan, E., Veit, A., Hua, Y., Estrin, D., & Shmatikov, V. (2020). How to backdoor federated learning. Proceedings of the 23rd International Conference on Artificial Intelligence and Statistics, 2938-2948.

7. Cheng, K., Fan, L., & Yang, Q. (2022). Split learning backdoor attacks and defenses. Proceedings of the 31st International Joint Conference on Artificial Intelligence, 2997-3004.

8. Shui, Y., Jin, R., Dou, Z., & Gao, Z. (2026). ProtoGuard-SL: Prototype Consistency Based Backdoor Defense for Vertical Split Learning. arXiv preprint arXiv:2604.03595.

9. Buterin, V. (2016). Ethereum: A next-generation smart contract and decentralized application platform.

10. Gao, Y., Zhang, Y., & Liu, Y. (2021). zk-SNARKs in machine learning: A survey. ACM Computing Surveys, 54(6), 1-35.

11. Tian, F. (2017). An agri-food supply chain traceability system for China based on RFID & blockchain technology. 2016 13th International Conference on Service Systems and Service Management, 1-6.

12. Hardjono, T., & Smith, N. (2021). Decentralized identity and verifiable credentials for IoT. IEEE Internet of Things Journal, 8(13), 10522-10535.

13. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, 1273-1282.

14. Cheng, H., Yu, Y., & Wang, X. (2023). A survey on blockchain-based machine learning: Architecture, methods, and applications. Journal of Systems Architecture, 137, 102850.

15. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., & Chandra, V. (2018). Federated learning with non-IID data. arXiv preprint arXiv:1806.00582.

16. Zhang, L., Xu, J., & Lin, F. (2022). Smart contract security: A survey. IEEE Access, 10, 26217-26241.

17. Kiayias, A., Russell, A., David, B., & Oliynykov, R. (2017). Ouroboros: A provably secure proof-of-stake blockchain protocol. Proceedings of the 37th Annual International Cryptology Conference, 357-388.

18. Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., ... & Yellick, J. (2018). Hyperledger Fabric: A distributed operating system for permissioned blockchains. Proceedings of the 13th EuroSys Conference, 1-15.

19. Jiang, Y., Li, J., & Wang, C. (2023). Trusted execution environments for secure federated learning: A survey. IEEE Transactions on Dependable and Secure Computing, 20(5), 3980-3997.

20. Xie, C., Huang, K., Chen, P. Y., & Li, B. (2020). DBA: Distributed backdoor attacks against federated learning. International Conference on Learning Representations.