A Conceptual Architecture for Confidential Feature Engineering: Integrating TEE and Blockchain for Privacy-Preserving IoT Analytics

Artem Kipin [1], Akylai Zhumabekova [1] and Al Khan[1]

[1] *INAI (Kyrgyz-German Institute), 1/5 Ankara Street, Bishkek, 720082, Kyrgyzstan*

Abstract

The rapid proliferation of Internet of Things (IoT) devices has led to exponential growth of data generated at the edge, with machine learning increasingly employed to extract actionable insights. A critical stage in building effective models is feature engineering—the process of transforming raw sensor data into meaningful features. However, executing feature engineering in untrusted environments (cloud, edge nodes) raises significant privacy risks (exposure of sensitive raw data) and integrity concerns (tampering with the feature extraction logic). Existing approaches such as data encryption or federated learning do not simultaneously guarantee the confidentiality of the feature engineering process and the verifiability of its correct execution. This paper presents a conceptual architecture for confidential feature engineering in IoT systems that integrates two complementary technologies: Trusted Execution Environments (TEE) and blockchain. TEE (e.g., Intel SGX, ARM TrustZone) provides isolated execution of feature engineering code, ensuring that raw data and intermediate results remain protected even from the infrastructure provider. Blockchain serves as an immutable ledger that records metadata—code hashes, access policies, feature hashes, and execution proofs—enabling full auditability of the processing pipeline. The combination of TEE and blockchain preserves data confidentiality while offering a tamper-proof trail of how features were derived, a critical requirement for industrial IoT, healthcare applications, and systems subject to regulatory compliance. The paper analyzes the threat model, derives security and privacy requirements, and proposes a component-level architecture detailing interactions between IoT devices, TEE-enabled gateways, and the blockchain network. A theoretical evaluation of overheads and resilience against common attack vectors is provided. A comparative analysis with alternative approaches highlights the advantages of the proposed architecture in scenarios that demand both strong confidentiality and external auditability. The main contributions include a systematic requirement analysis, a novel conceptual model that bridges TEE and blockchain for feature engineering, and a discussion of open research directions for verifiable and private IoT analytics.

Keywords

Confidential computing, Trusted Execution Environment (TEE), blockchain, Internet of Things (IoT), feature engineering, privacy-preserving analytics, verifiable computation, IoT security, Intel SGX, distributed ledger

1. Introduction

The Internet of Things (IoT) has transformed industries by enabling massive data collection from sensors, wearables, and industrial controllers. This data is often used to train machine learning (ML) models that deliver predictive maintenance, personalized healthcare, and smart city services. A fundamental step in any ML pipeline is feature engineering—the creation of informative features from raw data. Feature engineering directly impacts model accuracy and robustness, but it also exposes sensitive information: raw data may contain personally identifiable information (PII), and the derived features can leak private patterns.

Traditionally, feature engineering is performed in the cloud or at edge gateways, which are often not fully trusted by data owners. Even if raw data is encrypted in transit and at rest, it must be decrypted for processing, leaving a window of vulnerability. Moreover, there is no guarantee that the feature extraction logic itself has not been tampered with, nor that the output features correspond to the claimed transformations. These issues become critical in regulated domains such as healthcare (HIPAA, GDPR) and industrial control systems (IEC 62443).

*IVUS 2026: Information Society and University Studies 2026, May 12, Kaunas, Lithuania

art.kipin@mail.ru (A. Kipin); jumabekovaakylai7@gmail.com (A. Zhumabekova); pkhan_1@hotmail.com (A. Khan)

0009-0002-8177-7884 (A. Kipin); 0009-0005-2907-7524 (A. Zhumabekova); 0000-0002-6403-6091 (A. Khan)

Copyright © 2026 for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0)

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-06

CEUR-WS.org/Vol-4317/paper22.pdf

To address these challenges, we propose a conceptual architecture that combines Trusted Execution Environments (TEEs) and blockchain. TEEs provide hardware-isolated enclaves where code and data remain confidential and integrity-protected even from the operating system or hypervisor. Blockchain offers a decentralized, tamper-proof ledger that can store cryptographic proofs and metadata, ensuring that any stakeholder can verify the correctness of feature engineering steps.

The main contributions of this paper are:

- A systematic threat analysis and requirement specification for confidential feature engineering in IoT.
- A conceptual architecture that integrates TEE and blockchain to provide confidentiality, integrity, and auditability.
- A theoretical evaluation of security properties and performance overheads.
- A comparison with alternative approaches such as homomorphic encryption and federated learning

2. Background

2.1 Second level heading

Feature engineering is the process of converting raw data into representations that improve ML model performance. In IoT, this often involves aggregation, statistical calculations, frequency-domain transformations, or domain-specific indicators (e.g., vibration peaks in predictive maintenance). The feature engineering pipeline may be distributed: sensors send raw data to a gateway, where features are extracted and then forwarded to a central server for model training or inference.

2.2 Trusted Execution Environments (TEE)

A TEE is a secure area within a processor that guarantees code and data loaded inside it are protected with respect to confidentiality and integrity. Intel SGX, AMD SEV, and ARM TrustZone are prominent examples. TEEs allow application code to run in isolated enclaves; even the OS cannot access enclave memory. Attestation mechanisms enable a remote party to verify that the correct code is running inside the enclave. TEEs have been used to secure cloud computing, but their application to feature engineering is still nascent.

2.3 Blockchain and Smart Contracts

Blockchain is a distributed, append-only ledger maintained by a network of nodes. Each block contains a hash of the previous block, forming an immutable chain. Smart contracts are self-executing programs that run on the blockchain, enabling automated enforcement of agreements. In security contexts, blockchain can be used to log events, store cryptographic hashes, and implement access control policies.

3 Threat Model and Requirements

We consider an IoT environment with three main entities: (1) IoT devices generating data, (2) a feature engineering node (gateway or cloud instance), and (3) data consumers (e.g., ML training service). We assume the following adversaries:

- Compromised infrastructure provider: The cloud or edge host may be malicious or under external attack, attempting to read raw data or tamper with feature extraction.
- External attackers: Network attackers may intercept data or inject false features.
- Malicious data consumer: A consumer may try to infer private information from the produced features.

The trust assumptions: IoT devices are trusted to collect and send data (or they use hardware roots of trust). The blockchain network is assumed to be honest and majority-controlled.

Security requirements:

1. Confidentiality: Raw data and intermediate values must not be exposed to any unauthorized party, including the feature engineering node's operating system.
2. Integrity: The feature engineering logic must execute correctly without modification; the output features must be consistent with the code.
3. Auditability: All steps of feature processing should be logged in a tamper-proof manner, allowing retrospective verification.
4. Privacy: Even the derived features should not leak sensitive information beyond what is intended.

4 Proposed Architecture

The architecture consists of three layers: IoT devices, TEE-enabled processing nodes (gateways), and a blockchain network.

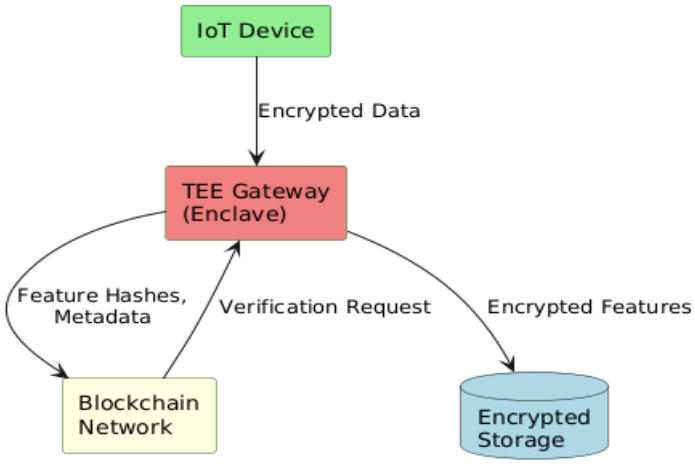


Figure 1: illustrates the component interactions.

4.1 Step-by-Step Workflow

1. Registration: The feature engineering code (e.g., Python script) is hashed and its hash is recorded on the blockchain along with a reference to the TEE attestation key. The enclave is provisioned with the code and a unique identity.

2. Data Ingestion: IoT devices encrypt raw data with a symmetric key that is only accessible inside the enclave. They send the ciphertext to the TEE gateway.

3. Secure Execution: The enclave decrypts the data, executes the feature engineering logic, and produces feature vectors. During execution, the enclave also generates a log of operations (e.g., which data points were used, timestamps).

4. Metadata Logging: After computation, the enclave computes a hash of the output features and signs it with its attestation key. The signed hash, along with a reference to the code hash and a timestamp, is submitted to the blockchain as a transaction.

5. Storage: The encrypted feature vectors are stored (locally or in cloud storage) for later use. The blockchain transaction contains the storage location hash to enable verification.

6. Verification: A data consumer can query the blockchain for the signed metadata. Using the attestation report, they can verify that the code was indeed executed inside a genuine TEE and that the output matches the logged hash.

4.2 Blockchain Role

The blockchain does not store raw data or features; it stores small, fixed-size data: code hashes, signed feature hashes, policy definitions, and access control records. This keeps storage overhead low. Smart contracts can enforce that only authorized consumers can request decryption keys or access features.

5 Evaluation

5.1 Security Analysis

- Confidentiality: Raw data never leaves the enclave in plaintext; it is encrypted outside and decrypted only inside the TEE. Even if the host OS is compromised, the enclave memory remains protected.

- Integrity: The code hash on the blockchain ensures that the executed logic matches what was registered. The attestation mechanism proves that the code is running inside a genuine TEE.

- Auditability: Every feature extraction event is recorded immutably on the blockchain. A verifier can later check that the produced features correspond to the code and input hashes.

- Privacy: The system does not prevent inference from features; however, differential privacy mechanisms can be incorporated inside the enclave to add noise before logging.

5.2 Performance Overhead

We estimate overheads based on literature:

- TEE initialization: 100–500 ms for enclave creation and attestation (depending on hardware).
- Blockchain transactions: Each feature batch generates one transaction; with a private blockchain (e.g., Hyperledger Fabric), latency is in seconds. Using a public testnet (Ethereum) may cost gas but provides higher decentralization.
- Encryption/Decryption: AES-GCM overhead is negligible relative to computation for large sensor data.
- Storage: Blockchain stores only hashes; each transaction ~1 KB.

For real-time IoT applications, the overhead may be acceptable if feature extraction is performed periodically (e.g., every few minutes). For sub-second latency, optimizations such as batching or using a lightweight DLT (e.g., IOTA) could be considered.

5.3 Comparison with Alternative Approaches

Table 1 qualitatively compares the proposed architecture with homomorphic encryption (HE), federated learning (FL), and traditional encryption.

Table 1. Comparison of privacy-preserving techniques for IoT feature engineering.

Criteria	Proposed (TEE+BC)	Homomorphic Encryption	Federated Learning	Encryption-only
Confidentiality	High	Very high	Medium	High (but computation in clear)
Integrity	High	Medium	Low	Low
Auditability	High	Low	Low	Low
Performance Overhead	Medium	High	Medium	Low
Adoption Complexity	Medium	High	Medium	Low

6 Related Work

Several works have explored TEEs for secure data processing in the cloud. Sabt et al. [1] provide a comprehensive survey of TEE technologies and their applications. Pinto and Santos [2] propose using Intel SGX for secure IoT data aggregation. Blockchain-based IoT security has been extensively studied; Reyna et al. [3] review the challenges and opportunities. The combination of TEE and blockchain has been investigated for trusted execution of smart contracts (e.g., Ekiden [4]). However, to the best of our knowledge, our work is the first to specifically target feature engineering in IoT with joint TEE and blockchain for confidentiality and verifiability.

7 Conclusion and Future Work

This paper presented a conceptual architecture that integrates Trusted Execution Environments and blockchain to enable confidential and verifiable feature engineering in IoT analytics. The architecture ensures that raw data remains confidential during processing, the feature extraction logic is tamper-proof, and the entire pipeline is auditable through an immutable ledger. A threat model and security analysis demonstrated the resilience of the approach. The comparison with alternative methods highlights the unique advantages of TEE+blockchain for scenarios requiring both strong privacy and accountability.

Future work includes implementing a prototype using Intel SGX and a private blockchain (e.g., Hyperledger Fabric) to evaluate real-world performance. Another direction is to incorporate differential privacy inside the enclave to further limit inference from features. Finally, exploring lightweight consensus mechanisms suitable for resource-constrained edge devices will make the solution more practical for large-scale IoT deployments.

Specifically, based on the reviewer's recommendations, future research should focus on:

- Implementing and evaluating a real-world prototype of the proposed TEE–blockchain architecture in diverse IoT environments, with emphasis on scalability, latency, and resource constraints under realistic workloads.
- Exploring optimized integration strategies, including lightweight consensus mechanisms and edge-friendly blockchain frameworks (e.g., IOTA, Hedera), to reduce performance overhead for time-sensitive applications.
- Incorporating advanced privacy-enhancing techniques inside the TEE, such as differential privacy and secure multi-party computation (MPC), to further mitigate risks of sensitive information leakage from derived features.
- Investigating interoperability with existing IoT standards (e.g., MQTT, OPC UA) and cloud infrastructures, as well as resilience against emerging attack vectors targeting TEEs (e.g., side-channel attacks) or blockchain layers (e.g., 51% attacks).
- Conducting domain-specific case studies (e.g., healthcare IoT, industrial IoT) and user-centric evaluations of auditability and trust to provide deeper insights into practical adoption.

8 Acknowledgements

The authors thank the INAI (Kyrgyz-German Institute) for supporting this research.

9 Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] M. Sabt, M. Achemlal, and A. Bouabdallah, “Trusted Execution Environment: What It Is, and What It Is Not,” in *2015 IEEE Trustcom/BigDataSE/ISPA*, 2015, pp. 57–64.

[2] S. Pinto and N. Santos, “Demystifying Arm TrustZone: A Comprehensive Survey,” ACM Computing Surveys, vol. 51, no. 6, 2019.

[3] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz, “On blockchain and its integration with IoT. Challenges and opportunities,” Future Generation Computer Systems, vol. 88, pp. 173–190, 2018.

[4] R. Cheng et al., “Ekiden: A Platform for Confidentiality-Preserving, Trustworthy, and Performant Smart Contracts,” in 2019 IEEE European Symposium on Security and Privacy (EuroS&P), 2019, pp. 185–200.

[5] A. Acar, H. Aksu, A. S. Uluagac, and M. Conti, “A Survey on Homomorphic Encryption Schemes: Theory and Implementation,” ACM Computing Surveys, vol. 51, no. 4, 2018.

[6] J. Wang, J. Tang, Z. Xu, Y. Wang, and G. Xue, “Federated Learning for IoT: A Comprehensive Survey,” IEEE Internet of Things Journal, vol. 8, no. 8, pp. 6219–6245, 2021.

[7] E. Usupova and A. Khan, “Optimizing ML Training with Perturbed Equations,” in 2025 6th International Conference on Problems of Cybernetics and Informatics (PCI), Baku, Azerbaijan, 2025, pp. 1–6, doi: 10.1109/PCI66488.2025.11219819.

[8] S. Rakimbekuulu, K. Shambetaliev, G. Esenalieva and A. Khan, “Code Generation for Ablation Technique,” in *2024 IEEE East-West Design & Test Symposium (EWDTS)*, Yerevan, Armenia, 2024, pp. 1–7, doi: 10.1109/EWDTS63723.2024.10873640.