

Dynamic Authentication and Privacy-Preserving Data Trading Scheme with Lattice in Blockchain-Enhanced Smart Logistics

Xue GAO, Ke-He WU, Xiao-Liang ZHANG, Wen-Chao CUI*

School of Control and Computer Engineering, North China Electric Power University, Beijing 102206, China
e-mail: gaoxue1026@ncepu.edu.cn, wkh@ncepu.edu.cn, zhanghino@126.com, cuiz@ncepu.edu.cn

Received: April 2026; accepted: August 2026

Abstract. Against the risks of privacy leakage and quantum computing attacks in smart logistics data trading, this paper proposes a post-quantum privacy-preserving data trading scheme built on consortium blockchain and federated learning. Three orthogonal core innovations are elaborated as follows:

- (1) A five-layer decoupled smart logistics data trading platform architecture is constructed. Different from existing single-layer blockchain trading frameworks, this work hierarchically divides logistics data workflows into perception, generation, transmission, trading platform and application layers, and introduces a blockchain oracle module to realize consistent synchronization between on-chain transaction records and off-chain IPFS raw cipher storage, solving the cross-layer data inconsistency defect of prior monolithic architectures.
- (2) A quantitative model parameter quality evaluation algorithm based on cosine similarity, together with a dual-weight dynamic incentive mechanism, is designed. Most existing federated data trading studies lack standardized quantitative quality metrics and static reward allocation rules; our method extracts gradient feature vectors from local FL models, calculates similarity against benchmark parameters to grade data quality, and distributes transaction revenue weighted by both model quality and service participation time to motivate sustained user contribution.
- (3) A complete lattice-based cryptographic suite including homomorphic proxy re-encryption and aggregate lattice signature is deployed for secure data circulation. Traditional pairing-based privacy schemes are vulnerable to quantum cracking, while standalone lattice encryption works cannot support dynamic cross-buyer ciphertext conversion; our lattice primitive supports homomorphic aggregation and one-time re-encryption transfer, realizing anti-quantum, flexible authorization of federated model trading ciphertexts. Theoretical security proofs and multi-dimensional simulation experiments verify that the proposed scheme achieves post-quantum privacy protection, accurate data quality quantification and high-efficiency dynamic logistics data transactions.

Key words: *Index Terms*—smart logistics, data trading, lattice cryptography, federated learning, consortium blockchain, post-quantum privacy preservation.

*Corresponding author.

1. Introduction

Data has become an important resource for economic and social development, and the key to leveraging the value of data lies in data circulation. Data transaction is a typical data circulation paradigm. Based on the role of data trading entities, there are currently three modes of data trading: unilateral data provision mode, data trading platform mode and data management system mode (Azcoitia and Laoutaris, 2022). This article focuses on the data trading platform mode.

In terms of existing data trading platform mechanisms, many studies have turned their attention to blockchain technology. In 2022, Khezzr *et al.* (2022) focused on the construction of a reputation system based on blockchain edge intelligence in the Industrial Internet of Things. Chen *et al.* (2022) expected to use blockchain and mathematical optimization technology to solve the problem of traditional centralized optimization and management solutions being incompatible with energy systems whose structures are increasingly decentralized and constantly changing. Hong *et al.* (2022) proposed a bargaining dynamic game model and a forward-looking reasoning method to solve the optimal choice problem in the carbon emissions trading process of industrial enterprises.

In the process of data transactions, many scholars have explored blockchain and federated learning to achieve an efficient data circulation model due to its excellent decentralization and natural data circulation characteristics. Xu *et al.* (2023) proposed an Industrial Internet of Things secure data sharing solution based on blockchain and federated learning for the massive data in the industry. Wang *et al.* (2022) considered cloud-edge communication and proposes a hierarchical federated learning incentive mechanism based on blockchain. Tu *et al.* (2022) considered the incentive mechanism of blockchain combined with federated learning from the perspectives of economics and game theory. Wang *et al.* (2022) focused on the consensus mechanism strategy of blockchain combined with federated learning, and proposes a platform-free strategy. Qi *et al.* (2022) also noticed the importance of reputation and proposed a blockchain-based federated learning with a reputation mechanism for high-quality model aggregation. Zhang *et al.* (2023) focused on resource allocation strategies in the data sharing process based on blockchain and federated learning. In view of the highly dispersed nature of logistics data, this article also uses the sharing model of blockchain and federated learning for data circulation and transactions.

However, the flow of private data has become an urgent problem to be solved. In particular, in 2016, the European Union introduced the General Data Protection Regulation (GDPR) policy to protect user and corporate privacy (GDPR, 2016). In 2021, Warnat-Herresthal *et al.* (2021) combined blockchain, federated learning and privacy protection strategies, proposes swarm learning, and applies this method in clinical practice. In 2019, Lu *et al.* (2019) used differential privacy technology to achieve privacy protection based on federated learning and blockchain shared data. Zhou *et al.* (2022) propose a secure and trusted federated data sharing based on blockchain in Industrial Internet of Things. Yang *et al.* (2022) aimed to design an privacy-preserved credit data sharing model to guarantee the secure storage and sharing of credit information in a decentralization environment.

Chen *et al.* (2023) hoped to achieve privacy protection and traceable federated learning in industrial applications. Mazzocca *et al.* (2023) had designed a model of trusted federated learning as a service. Li and Shi (2022), Li *et al.* (2021, 2022a), Li *et al.* (2022b), Li (2025), Li and Shi (2024), Li (2024a, 2024b) used blockchain and cryptography technologies, such as homomorphic encryption, to achieve privacy protection during data sharing. Lo (2022) proposed a trustworthy AI blockchain data circulation architecture. Yang *et al.* (2022) proposed a blockchain transaction model based on trusted federated learning from the perspective of Byzantine attacks.

However, although the above blockchain-federated learning data sharing solutions have achieved preliminary progress in general industrial data trading scenarios, three prominent universal research limitations still exist. More importantly, these universal defects are further amplified when applied to smart logistics scenarios due to the unique fragmented, heterogeneous, time-sensitive characteristics of logistics business data, and existing schemes cannot adapt to the personalized industrial requirements of logistics data circulation. The three core targeted research gaps are analysed as follows:

- (1) **Research gap 1:** Most existing data trading frameworks lack standardized quantitative evaluation methods for data source quality, and only adopt rough qualitative discrimination standards for shared data. Smart logistics data is collected by massive scattered terminals including trucks, warehouses, distribution stations and express outlets, showing obvious fragmentation and heterogeneity: different logistics operators generate order data, temperature sensor data, vehicle trajectory data with inconsistent feature dimensions and data distribution. Traditional unified quality evaluation rules designed for standardized industrial datasets cannot classify model quality based on heterogeneous logistics feature vectors. If directly deployed in logistics data markets, high-value trajectory models and low-quality noisy order models cannot be quantitatively distinguished, resulting in low-efficiency matching between data buyers and sellers in logistics transactions.
- (2) **Research gap 2:** Mainstream data trading incentive mechanisms only set static reward distribution rules based on single participation indicators, without comprehensive weighting of multiple service contribution dimensions. Logistics data providers include large logistics platforms, small individual couriers and regional warehousing enterprises, with highly differentiated data output quality and continuous service cycles. General single-factor incentive mechanisms only calculate revenue based on data upload frequency, ignoring the core demand of logistics scenarios to reward high-quality fragmented sensing data. Small and medium-sized logistics merchants who provide high-precision temperature/trajectory model parameters cannot obtain matching transaction returns, leading to insufficient long-term participation willingness and insufficient supply of high-quality heterogeneous logistics data on the trading platform.
- (3) **Research gap 3:** Most privacy-preserving data circulation cryptography relies on bilinear pairing primitives vulnerable to quantum attacks, and cannot support dynamic re-encryption authorization for cross-buyer data trading. Logistics data transactions involve frequent temporary authorization: the same batch of heterogeneous vehicle/warehouse model parameters need to be shared with multiple logistics demanders

such as route optimization platforms and warehouse scheduling systems. Traditional pairing-based homomorphic encryption schemes are twofold incompatible with logistics scenarios: on the one hand, pairing cryptography faces quantum cracking risks threatening sensitive delivery address and cargo temperature data; on the other hand, static encryption cannot flexibly re-authorize ciphertexts to different buyers, requiring repeated plaintext re-upload of fragmented logistics model parameters and bringing serious privacy leakage risks.

To address quantum attack risks, unquantifiable logistics data quality, and on/off-chain data inconsistency defects of existing blockchain-federated learning data trading solutions, this paper proposes a lattice-based privacy-preserving dynamic data trading architecture for smart logistics. Three mutually independent, non-overlapping core contributions are summarized below, with clear technical breakthroughs against state-of-the-art literature respectively:

- (1) Layered consortium blockchain-federated learning data trading platform architecture. We design a five-layer decoupled logistics data trading system composed of data perception layer, data generation layer, network transmission layer, blockchain trading platform layer and scenario application layer. Compared with prior blockchain data trading architectures which adopt undifferentiated monolithic structure without separation of raw data collection, model training and on-chain transaction management, our layered design isolates heterogeneous logistics service modules. A dedicated blockchain oracle mechanism is embedded to synchronize off-chain IPFS ciphertext storage and on-chain transaction ledgers, completely resolving the data consistency gap ignored by single-layer platform designs. This contribution solely focuses on system architectural innovation without mixing cryptographic or incentive algorithm designs.
- (2) Cosine-similarity model quality quantification & time-quality joint incentive mechanism. Existing federated learning data trading works fail to provide computable quantitative standards to evaluate the quality of uploaded model parameters, and their incentive strategies only rely on single participation time without data quality weighting. We first extract gradient feature vectors from local federated models, compute cosine similarity relative to a global benchmark model to generate standardized quality scores for each data seller's model. On this basis, a dual-weight revenue distribution formula is proposed, which allocates transaction profits according to both model quality and continuous service time to improve long-term user participation enthusiasm. This innovation independently targets data quality assessment and incentive algorithm, excluding architectural and post-quantum cryptography content.
- (3) Lattice homomorphic proxy re-encryption and aggregate lattice signature suite for post-quantum circulation. Nearly all mainstream logistics privacy schemes rely on bilinear pairings, which are vulnerable to quantum computer attacks; existing lattice cryptography research only supports static ciphertext storage and lacks dynamic re-encryption authorization for data trading. We integrate Lattice-based Homomorphic Proxy Re-encryption (LHPRE) and lattice aggregate signature (LHS) algorithms. The

LHPRE primitive supports homomorphic aggregation of FL model ciphertexts and one-time re-encryption conversion for different buyers, while the LHS algorithm realizes batch integrity verification of trading records. The entire cryptographic stack is post-quantum secure under the LWE hardness assumption, filling the gap of anti-quantum dynamic data circulation tools for logistics data markets. This contribution is purely cryptographic innovation with no repeated architectural or incentive descriptions.

The remainder of this paper is organized as follows. Section 2 introduces lattice cryptography and cosine similarity preliminaries. Section 3 formalizes system entities, security assumptions and design objectives. Section 4 details the five-phase full scheme workflow covering initialization, FL training, quality evaluation, trading and dynamic tracking. Section 5 conducts rigorous LWE-based security proofs and evaluates computational/communication overhead via trace-driven logistics simulations. Section 6 concludes this work and outlines future research directions.

2. Preliminary

2.1. Lattice-Based Homomorphic Proxy Re-Encryption Scheme

Lattice-based homomorphic proxy re-encryption scheme (LHPRE) (Luo *et al.*, 2020; Li *et al.*, 2023) includes six algorithms: **Setup**, **KeyGen**, **Enc**, **Dec**, **ReKeyGen**, **ReEnc**.

Setup($1^\lambda, 1^d, 1^{\mathcal{T}}$): Input security parameter λ , output $pp, mpk := (\mathbf{A}, \mathbf{A}_1, \mathbf{B}, \mathbf{u})$ and $msk := (\mathbf{S})$.

KeyGen(msk, id): Given msk and identity id , output a secret key $sk_{id}: \mathbf{s} = (1, -\mathbf{z}^t) \in \mathbb{Z}^{2m+1}$ and the public key $pk_{id}: \mathbf{A}_{id} = [\mathbf{u} \parallel \mathbf{H}]^t \in \mathbb{Z}_q^{(2m+1) \times n}$.

Enc(mpk, id, μ): Input μ , computes $\mathbf{C} = \mathbf{A}_{id}\mathbf{R} + 2\mathbf{E} + \mu\mathbf{G}$.

Dec($\mathbf{C}, id, sk_{id}$): Let $\mathbf{c}$ be the first column of $\mathbf{C}$. Get $\mu = \langle \mathbf{s}, \mathbf{c} \rangle \bmod q \bmod 2$.

ReKeyGen($sk_{id}, id, id', pk_{id'}$): Given sk_{id} , and $pk_{id'}$, output the re-encryption key $sk_{id \rightarrow id'}: \mathbf{P}_{id \rightarrow id'} = \mathbf{A}_{id'}\mathbf{R}' + 2\mathbf{E}' + \begin{bmatrix} \mathbf{s}\mathbf{G} \\ \mathbf{0} \end{bmatrix} \in \mathbb{Z}_q^{(2m+1) \times N}$.

ReEnc($sk_{id \rightarrow id'}, id', (id, \mathbf{C})$): Given $sk_{id \rightarrow id'}$, and $\mathbf{C}$, output $\mathbf{C}_{id \rightarrow id'} := \mathbf{P}_{id \rightarrow id'}\mathbf{G}^{-1}(\mathbf{C})$.

The algorithm satisfies homomorphism, that is $\mathbf{Enc}(\mu_1 + \mu_2) = \mathbf{Enc}(\mu_1) + \mathbf{Enc}(\mu_2)$ and $\mathbf{Enc}(\mu_1 \cdot \mu_2) = \mathbf{Enc}(\mu_1) \cdot \mathbf{G}^{-1}(\mathbf{Enc}(\mu_2))$. Its correctness is shown in Luo *et al.* (2020), Li *et al.* (2023), which it has been proved that the algorithm meets CPA security in semantic security.

2.2. Lattice-Based Homomorphic Signature Scheme

Lattice-based homomorphic signatures scheme (LHS) (Gorbunov *et al.*, 2015) includes four algorithms: **Setup**, **KeyGen**, **Sign**, **Verify**.

Setup($1^\lambda, 1^N$): Choose $v_1, v_2, \dots, v_N \leftarrow \mathbb{Z}_q^n$, output $prms = (v_1, v_2, \dots, v_N)$.

KeyGen(id): Given identity id , output the public and private keys $pk_{id,sig} = \mathbf{A}_{id}$, $sk_{id,sig} = \mathbf{td}_{id} \leftarrow \text{TrapGen}(1^n, 1^m, q)$.

Sign($id, pk_{id,sig}$): Given identity id and the public key $pk_{id,sig}$, output the signature $\sigma_i = \text{SamPre}(\mathbf{A}_i, v_i - \mu \cdot \mathbf{G}, \text{td}_i)$.

Verify($id, pk_{id,sig}, \sigma_i$): Given identity id , the public key $pk_{id,sig}$ and the signature σ_i , output accept or reject by checking $f(\sigma_i) = \text{Process}(\mathbf{G})$ or not.

The algorithm also satisfies homomorphism: **Sign**($\mu_1 + \mu_2$) = **Sign**(μ_1) + **Sign**(μ_2). Its correctness is shown in Gorbunov *et al.* (2015), which has proved that the algorithm meets semantic security in LWE assumption.

2.3. Cosine Similarity Measures

Evaluate the similarity between two vectors by calculating the cosine value of their included angle (Ye, 2011). Cosine similarity draws vectors into vector space based on coordinate values, such as the most common two-dimensional space. Cosine similarity focuses on the angular relationship between vectors, not on their absolute size. The range of values is $[-1, 1]$. When measuring the similarity of a pair of texts, when the text content is similar, if word frequency or word vectors are used as features, their Euclidean distance in the feature space may be large. However, the angle between them may be small, resulting in high similarity. Here, this article only focuses on the direction between vectors, and the direction is one direction, so the value range is $[0, 1]$. The closer it is to 1, the more similar the feature vectors of the two parameters are, and vice versa, the less similar they are.

3. Model and Design Goal

In this section, we formalize the data trading architecture, system model, security model and design goal. The parameters and explanations used in this paper are shown in the Table 1 below.

All abbreviations defined in this table are the exclusive standard notation adopted across the full manuscript; alternate variant shorthands are prohibited.

3.1. Data Trading Architecture and System Model

There are many ways to trade data now. In particular, this article focuses on the data trading platform model, whose advantage is that buyers and sellers can conduct open and free transactions bilaterally, and the platform manages their data transaction processes and related assets. This article designs a smart logistics data trading platform architecture, including five layers: trading data perception layer, trading data generation layer, network layer, data trading platform layer, data trading application layer, as shown in the Fig. 1.

- **Trading data perception layer:** This layer contains many sensors or terminal devices in the logistics field (such as position sensors, RFID, radar, GPS, temperature sensors, etc.) and is mainly used to collect raw data.
- **Trading data generation layer:** This layer mainly processes raw data and generates transaction data, such as training the original data set to obtain AI models or model parameters.

Table 1
Some used symbols and explanations.

Parameters	Description
λ	Security parameter.
<i>DTP</i>	Data trading platform.
<i>IPFS</i>	InterPlanetary File System.
<i>DB</i>	Data buyer.
<i>DS</i>	Data seller.
<i>pp</i>	System public parameters.
<i>pid</i>	User unique pseudonym identifier.
<i>mpk</i>	Master public key.
<i>msk</i>	Master private key.
<i>C</i>	Ciphertext.
σ	Signature.
<i>kw</i>	Trading data keyword tag.
A, R, E	Lattice public matrix, random blinding matrix, noise matrix.
<i>pk, sk</i>	Public and private key.
<i>s, z</i>	User secret vector, auxiliary random vector.
<i>q</i>	Lattice prime modulus (scalar).
<i>m, n, H</i>	Lattice dimension parameters.
<i>H, H₁</i>	Security hash function.

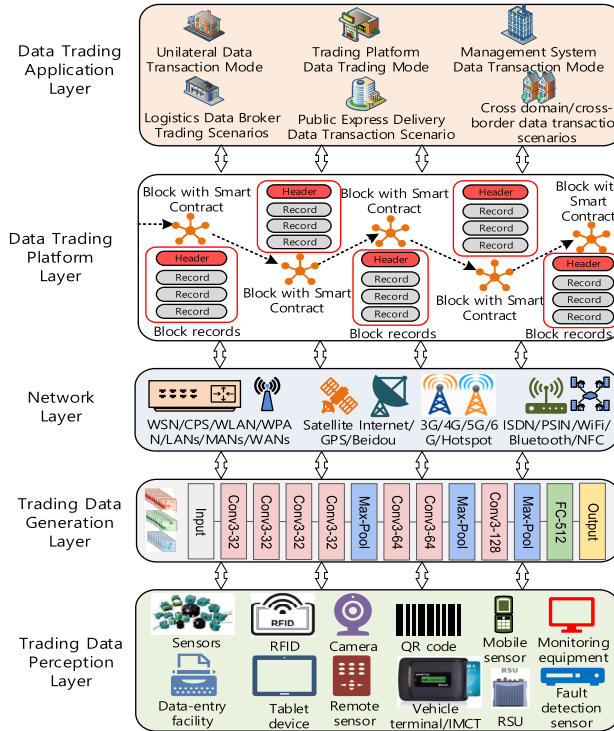


Fig. 1. Data trading architecture in smart logistics.

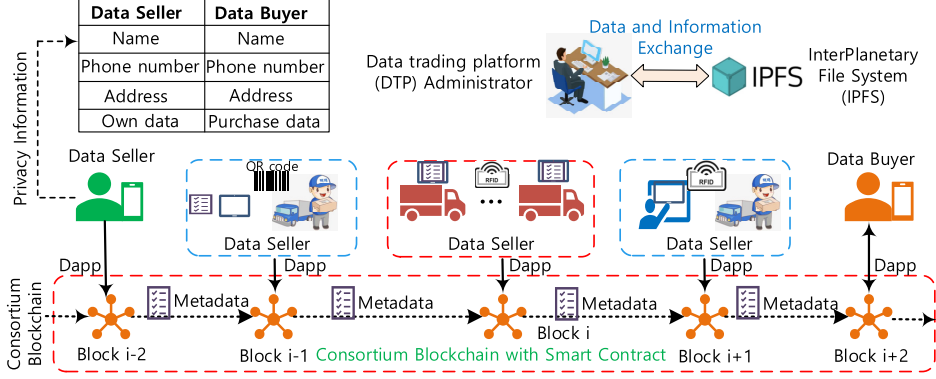


Fig. 2. System model.

- **Network layer:** This layer mainly provides network connection services. The connection methods can be wireless LAN, WSN, WPAN, GPRS, 3G, 4G, 5G, 6G, satellite network, quantum network, ISDN, PSIN, etc.
- **Data trading platform layer:** This layer is the trading platform layer, which is based on the alliance chain and smart contracts and provides trading platform services to buyers and sellers. Buyers or sellers can find potential customers on the platform and benefit from it.
- **Data trading application layer:** This layer has a large number of smart logistics scenarios and their applications, including freight Internet of Things data transactions, smart warehousing site selection data transactions, smart warehousing management data transactions, smart distribution data transactions, smart park freight management data transactions, etc.

Based on this architecture, a flexible and dynamic privacy-preserving data trading scheme with lattice homomorphism and federated learning in blockchain-enhanced smart logistics is designed, whose system model is shown in Fig. 2. In the model, there are four entities: data trading platform, InterPlanetary File System (IPFS), data buyer, data seller.

- **Data trading platform (DTP):** The *DTP* is responsible for providing data transaction services to registered users. Users with data selling needs and data buying needs can find potential customers on this platform. The platform has an administrator who is responsible for user registration, key generation and distribution, smooth transaction process, data quality measurement and negotiated pricing, model parameter aggregation and re-encryption, incentive calculation and issuance, feedback adjustment operations. It is worth noting that dynamic access control and authentication are also completed here.
- **InterPlanetary File System (IPFS):** The IPFS, as a distributed storage system, provides storage functions. Both blockchain data and user data can be stored here.
- **Data buyer (DB):** The DB has the need to buy data and can find the required data on the data trading platform and conduct transactions.

- **Data seller (DS):** The DS has the need to sell data and can find potential customers on the data trading platform and conduct transactions to obtain profits. Data sellers have a series of terminal devices that can sense and collect large amounts of data in real time.

3.2. Formal Trust and Threat Model

This article proposes a privacy-preserving logistics data transaction scheme, setting the DTP to be credible and other participants to be semi-credible (honest but curious). In addition, it is assumed that there is a malicious attacker outside the system, hoping to obtain some useful information through open eavesdropping, interception and retransmission, conspiracy attacks and other attack methods.

- (1) Entity Trust Classification. We first strictly define the trust level of each participant in the consortium blockchain logistics data trading system:
 - 1) Data Trading Platform (DTP): Regulated trusted participant under consortium multi-party governance (core trust assumption with industrial & cryptographic justifications provided below).
 - 2) Data Seller (DS, logistics merchants/warehouses) & Data Buyer (DB, logistics demand parties): Honest-but-curious (HbC, semi-trusted). They strictly follow protocol workflows but passively attempt to infer others' private model data from ciphertext transcripts.
 - 3) External Active Adversary $\mathcal{A}_{ext}$: Fully malicious untrusted attacker. Controls public communication channels, executes eavesdropping, replay, man-in-the-middle transmission attacks, but cannot break LWE lattice hardness assumptions.

Justification for Trusted DTP Assumption in Consortium Blockchain Logistics Scenarios. The DTP is designated as a trusted node only for permissioned Hyperledger Fabric consortium deployments, supported by two layers of guarantees: Institutional consortium governance constraint and Built-in cryptographic safeguards limiting DTP data leakage capability. The DTP operator is jointly nominated and supervised by multiple independent logistics enterprises, third-party regulatory auditors and equipment manufacturers that form the consortium committee. All core platform logic (user registration, re-encryption, pricing, incentive settlement) is hard-coded into open-source, auditable Fabric chaincode. Any platform operation record is timestamped and aggregated to multi-peer blocks; the consortium can launch joint arbitration and suspend DTP permissions if abnormal behaviours are detected. There is no single organization controlling the platform independently, eliminating unilateral malicious operation incentives. Even if the DTP is honest-but-curious or partially compromised, a series of hard-coded chain rules and lattice cryptographic restrictions prevent it from extracting plaintext private logistics data or tampering trading parameters arbitrarily, as detailed in Section 2 below. The trusted assumption only means the DTP will not actively launch large-scale malicious attacks; it does not grant the platform unrestricted access to plaintext sensitive information.

- (2) Technical Safeguards Against Curious / Compromised DTP. To eliminate single-point risks caused by DTP's multi-core administrative authorities (registration, key generation, proxy re-encryption, quality scoring, incentive allocation), we embed five

mandatory protection mechanisms into the smart contract and lattice cryptography workflow:

Safeguard 1: Distributed master key split for lattice cryptography. The global lattice master public key mpk is jointly generated via multi-peer distributed setup across consortium nodes, not solely produced by DTP. All user individual trapdoor secret keys td_{id} are locally derived by each DS/DB after receiving shared public parameters; the DTP never stores or accesses any user's complete private trapdoor. Even if DTP is compromised, it cannot obtain the decryption secret required to recover plaintext FL model parameters from ciphertexts.

Safeguard 2: Verifiable cosine similarity quality evaluation chaincode. All gradient feature vector extraction and cosine similarity CS_i calculation logic is deployed as open, deterministic on-chain smart contracts. The DTP only triggers the computation task and cannot modify the feature extraction algorithm or similarity formula. Any DS can upload test benchmark model parameters to the chain and verify the DTP's quality scoring result against the fixed algorithm; tampered quality scores will be identified and logged as arbitration evidence.

Safeguard 3: Zero-plaintext proxy re-encryption constraint. The LHPRE re-encryption key $sk_{DTP \rightarrow DB}$ is generated by combining DB's public lattice key, without exposing DS's original plaintext model data to DTP. The DTP only manipulates blinded LWE ciphertext groups during re-encryption, and the lattice noise term permanently masks plaintext information. Even a curious DTP cannot strip LWE noise to recover raw logistics model values during re-encryption processing.

Safeguard 4: Multi-party arbitration for incentive & pricing parameters. The weight coefficients $w_1 = 0.6$, $w_2 = 0.4$ of the incentive formula, as well as unit data pricing rules, are written into immutable genesis block chaincode. The DTP cannot unilaterally adjust reward distribution standards; any parameter modification requires multi-consortium peer voting with majority signature approval. All historical settlement records are archived for audit to prevent DTP's malicious profit manipulation.

Safeguard 5: Separation of oracle write & read authority. The oracle module responsible for IPFS on/off-chain synchronization runs on independent Fabric peer nodes isolated from the DTP service backend. DTP only sends ciphertext upload requests to the oracle, and cannot directly modify or delete off-chain IPFS storage files or tamper with the on-chain hash binding records. All synchronization logs are independently stored by oracle nodes for cross-checking against DTP operation traces.

- (3) Formal Threat Capability Bound Definition. 1) HbC DS/DB: Can store all received ciphertext, signature and transaction metadata; cannot forge valid lattice signatures or modify on-chain records without BLS/LHS signature failure; cannot reconstruct plaintext even by collating multiple transaction transcripts (proven in Theorem 3). 2) Curious/compromised DTP: Can execute all authorized platform workflow operations, but is blocked by the five above safeguards from recovering user plaintext data or arbitrarily altering trading calculation parameters. 3) External Adversary $\mathcal{A}_{ext}$: Can intercept all public channel ciphertext packets, replay old transactions, inject forged transmission requests; cannot solve LWE lattice problems to break ciphertext semantic security.

- (4) **Design Goal Supplement.** Based on the clarified trust model, we add a targeted design goal: The system shall resist privacy leakage and parameter tampering risks even when the DTP node is honest-but-curious or partially compromised, via distributed key generation, verifiable on-chain algorithms and multi-party consortium arbitration mechanisms.

3.3. Design Goal

Based on the data trading platform model, this paper designs a privacy protection model suitable for smart logistics data transactions. Based on this, the goals to be achieved by this plan are as follows:

- (1) **Privacy protection.** The private data involved in this solution includes original data, transaction data, user identity information, etc. One of our purposes is to protect these private data from being leaked.
- (2) **Flexible and dynamic authentication and transactions.** This solution needs to achieve flexible and dynamic data transactions, especially for flexible re-encryption and dynamic updating of the ciphertext of transaction data.
- (3) **Feasible incentive mechanism and reliable data quality assessment.** This solution is designed based on the trading platform. Implementing a feasible incentive mechanism to stimulate user enthusiasm for participation and reliable data evaluation methods need to be considered.

4. Privacy-Preserving Data Trading Scheme

4.1. Design Rationale

Logistics data generally refers to the general term for knowledge, materials, images, data, and files of various logistics activities. It has the characteristics of various types, complex sources, large amount of information, and fast update speed. Data trading can not only better leverage the value of the data itself, but also better predict market demands and changes, optimize transportation routes, and improve warehousing efficiency. Most of the current smart logistics data transaction architecture is based on blockchain platforms, and shared privacy protection research is mostly based on cryptography and privacy computing technology. However, problems such as the uneven quality of transaction data, imperfect incentive mechanisms, inconsistencies on and off the chain, and quantum computer attacks have still not been solved well. Based on this, this article considers that the above problems can be solved well based on the lattice homomorphic proxy encryption algorithm, blockchain and privacy computing.

This article hopes to build an open, secure, and incentive-based data transaction paradigm. Different from traditional data transactions, the transaction data in this article is logistics data, and the embodied form of transaction data is the model or parameters after model training, rather than the original data or data set. On the one hand, this can

improve the efficiency and accuracy of data sharing, and on the other hand, it can also protect the identity privacy of data users to the greatest extent. The specific structure and detailed description of the scheme are as follows.

4.2. Detailed Construction

The proposed privacy-preserving data trading scheme with lattice homomorphism and federated learning in blockchain-enhanced smart logistics in this article contains four phases: initialization phase, model training, parameter upload and parameter quality assessment phase, model parameter data transaction phase, trading data update and tracking phase.

(1) **Initialization phase:** The initialization phase consists of two parts: public parameters generation, user registration and key generation.

1) **Public parameters generation.** The *DTP* randomly chooses $v_1, v_2, \dots, v_N \leftarrow \mathbb{Z}_q^n$ for the signature algorithm LHS and inputs the security parameter λ to the LHPRE.Setup algorithm, outputs $pp := (n, m, q, N, \chi, \sigma, v_1, v_2, \dots, v_N)$, $mpk := (\mathbf{A}, \mathbf{A}_1, \mathbf{B}, \mathbf{u})$ and $msk := (\mathbf{S})$.

2) **User registration and key generation.** Each user (*DB* and *DS*) with *id* needs to register and obtains pseudonym *pid*, executes the LHPRE.KeyGen and LHS.KeyGen algorithms. Given $\mathbf{A} \in \mathbb{Z}_q^{n \times m}$, $\mathbf{A}_1 + H(id)\mathbf{B} \in \mathbb{Z}_q^{n \times m}$, a basis $\mathbf{S}$ for lattice $\Lambda_q^\perp(\mathbf{A})$, $\mathbf{u} \in \mathbb{Z}_q^n$, $\text{td}_{id} \leftarrow \text{TrapGen}(1^n, 1^m, q)$ and $\sigma > O(\sqrt{\log n}) \cdot \omega(\sqrt{\log(2m)})$, output a secret key $sk_{id}: \mathbf{s} = (1, -\mathbf{z}^t) \in \mathbb{Z}^{2m+1}$ and the corresponding public key $pk_{id}: \mathbf{A}_{id} = [\mathbf{u} \parallel \mathbf{H}]^t \in \mathbb{Z}_q^{(2m+1) \times n}$, and the public and private keys of signature algorithm: $pk_{id, sig} = \mathbf{A}_{id}$, $sk_{id, sig} = \text{td}_{id}$ for identity *id*.

(2) **Model training, parameter upload and parameter quality assessment phase:** This phase mainly includes the collection of original data, model training, encrypted upload of model parameters, verification, model quality assessment and off-chain batch storage.

1) **Collection of raw data.** Within a local domain, raw data is collected by the associated data seller (usually the data owner). These data may involve different aspects, and data sellers need to classify, preprocess and clean these data.

2) **Model training.** The data seller DS_i performs the federated learning model training operation and obtains the transaction data, that is, the model parameter MP_i . The DS_i can collaborate with other participants for training, or they can perform federated collaboration with the *DTP* administrator to finally obtain transaction data. Note that the transaction data (model parameters) here are parameters of the federated model *FedM*.

3) **Encrypted upload of model parameters.** The DS_i encrypts MP_i by LHPRE.Enc algorithm with the *DTP*'s public key to obtain $\mathbf{C}_i = \mathbf{A}_{DTP}\mathbf{R}_i + 2\mathbf{E}_i + MP_i \cdot \mathbf{G}$, $\sigma_i = \text{SamPre}(\mathbf{A}_i, v_i - \mathbf{C}_i \cdot \mathbf{G}, \text{td}_i)$ and $M_i = (\mathbf{C}_i, \sigma_i, \text{pid}_i, kw_i, t_i)$, where kw_i denotes the keywords of the model parameter and t_i represents the timestamp, which is used to resist replay attacks. Then, the DS_i sends M_i to the *DTP*.

- 4) **Verification.** After receiving the message, the *DTP* needs to verify the legitimacy of the identity. In order to improve efficiency, multiple signatures are aggregated on the same stage and then verified as a whole.
- 5) **Model quality assessment.** Assessment of data quality is an important task, especially for transactional data. This article uses the cosine similarity measurement method to evaluate the similarity between model parameters, and then obtain the quality *quali_td* of transaction data. For a more detailed introduction, see Section 4.3.
- 6) **Off-chain batch storage & Oracle bidirectional synchronization mechanism.** The *DTP* is stored off-chain in IPFS based on the type and quality classification of transaction data. Note that the data stored here is still in ciphertext state. Finally, the hash address *Torrent.hash* returned by IPFS together with *kw* and *price* are stored in the blockchain in the form of transaction records, where *price* is the price of the data, determined by the platform based on data quality and demand. Blockchain oracle technology (Meng *et al.*, 2021) is used here to preserve consistent information on and off the chain. For a more detailed introduction, see Section 4.3. After passing signature verification and quality scoring, the *DTP* invokes a dedicated logistics data blockchain oracle to complete bidirectional data synchronization between off-chain IPFS distributed storage and consortium blockchain ledger, realizing persistent consistency of ciphertext storage addresses and on-chain transaction records. The custom logistics oracle deployed on Fabric peers contains two core sub-modules: Write Synchronization Submodule: Responsible for pushing uploaded ciphertext batches to IPFS, computing unique content hash, and writing hash metadata to on-chain blocks; Consistency Check & Error Correction Submodule: Periodically pulls IPFS file metadata and cross-compares with on-chain stored hash values; triggers automatic repair when hash mismatch or file missing anomalies are detected.

Complete Real-Time Synchronization Workflow (Executed after DS cipher submission)). Step 1: The oracle receives validated ciphertext batch $M_i = (C_i, \sigma_i, pid_i, kw_i, t_i)$ from *DTP* smart contract. Step 2: Oracle calls IPFS client API to batch store ciphertext tuples into distributed off-chain storage, and returns immutable IPFS content-addressable hash *Torrent.hash* after upload completion. Step 3: The oracle packages a standardized on-chain synchronization transaction, which includes: DS pseudonym *pid_i*, data keyword *kw_i*, transaction timestamp *t_i*, quality score *quali_{id}*, IPFS storage hash *Torrent.hash*, negotiated transaction price *price*. Step 4: The synchronization transaction is broadcast to Fabric Raft consensus peers, packed into blocks and permanently written to the on-chain transaction ledger. Only lightweight metadata (hash, identity, price) is recorded on-chain; heavy ciphertext raw data is fully stored off-chain on IPFS to reduce blockchain storage overhead. Step 5: After block confirmation, the oracle writes a synchronization success flag back to the IPFS file's extended metadata, marking the ciphertext batch as "on-chain bound" to complete one round of consistent write synchronization.

The oracle executes a background consistency polling task every 300 seconds to eliminate storage inconsistency risks caused by network jitter, IPFS node downtime,

or incomplete uploads. Three typical fault handling rules are defined: IPFS file loss anomaly: When the oracle cannot retrieve the ciphertext file corresponding to an on-chain recorded *Torrent.hash* from the IPFS cluster, it automatically pulls the backup ciphertext copy cached temporarily by DTP, re-uploads to IPFS, updates the new hash value to a supplementary on-chain correction transaction, and logs the repair event in the oracle audit table. Hash mismatch anomaly: If the ciphertext file read from IPFS generates a hash value inconsistent with the on-chain recorded *Torrent.hash*, the oracle determines the ciphertext has been tampered off-chain; it triggers a transaction invalidation notification on-chain, freezes the corresponding data seller's transaction qualification, and retransmits the original verified ciphertext batch to regenerate a valid IPFS hash for ledger alignment. Asynchronous upload delay anomaly: If the oracle fails to receive the IPFS upload completion callback within a 10-second timeout window, it suspends the on-chain record writing process, retries the IPFS storage interface three times sequentially, and only commits the synchronization transaction after obtaining a valid hash to avoid empty invalid ledger entries.

- (3) **Model parameter data trading phase:** This phase mainly includes the transaction requests, confirmation and payment, re-encryption and distribution, data reading, and income distribution.

- 1) **Transaction requests.** The DB_j hopes to purchase some high-quality federated learning model parameters, taking smart logistics address recognition as an example. The DB_j retrieves relevant records on the platform, selects the data who wish to purchase based on keyword descriptions, prices, etc., and sends a purchase request to the DTP.
- 2) **Dynamic authentication.** After receiving a dynamic model transaction request, the DTP performs dynamic authentication operations. This authentication adopts a role-based access control strategy and dynamically authenticates based on dynamic transaction requests, using a one-time authentication approach.

This scheme adopts one-time session dynamic authentication with role-based access control (RBAC). Each data trading session generates an exclusive short-lived authentication credential bound to the user pseudonym, trading keyword and time window. We formalize the full lifecycle including credential generation, verification, policy update, expiration revocation and failure handling as Algorithm 1.

- 1) Credential lifetime: 1 trading hour (3600s); after expiration, the credential is automatically invalid and cannot pass verification.
- 2) Static base access policy: DS credentials allow data upload; DB credentials only allow data download requests; cross-role operation triggers authentication failure.
- 3) Revocation ledger: A dedicated on-chain 'AuthRevokeList' world state table records all invalidated pseudonym IDs and expired credential timestamps.

- 3) **Confirmation and payment.** The DTP confirms the data that needs to be purchased and sends a payment request to the DB_j . The DB_j completes confirmation and payment, and returns completed payment information.
- 4) **Re-encryption and distribution.** Once the DB_j completes the payment, the DTP generates the re-encryption key $sk_{DTP \rightarrow j}$: $\mathbf{P}_{DTP \rightarrow j} = \mathbf{A}_j \mathbf{R}_j + 2\mathbf{E}_j +$

Algorithm 1: Dynamic One-Time Authentication Lifecycle Algorithm

```

1 Global: On-chain access policy table PolicyTable, revocation
  ledger AuthRevokeList, lattice hash function H, system
  timestamp  $t_{sys}$ 
2 Input: User pseudonym pid, user role  $role \in \{DS, DB\}$ , target
  trading keyword kw, session start time  $t_0$ 
3 Output: Valid dynamic auth credential AuthCred /
  Authentication success flag / Revocation record
  // Step 1: Dynamic Credential Generation (User Local
  Execution)
4 User retrieves its long-term lattice signature secret key  $sk_{id,sig}$ 
  from local storage;
5 Compute session binding random nonce  $r \xleftarrow{\$} \mathbb{Z}_q$ ;
6 Encode session context vector  $Ctx = (pid, role, kw, t_0, t_0 + 3600)$ 
  (3600s lifetime);
7 Compute session digest  $h_{ctx} = H(Ctx || r)$ ;
8 5: Generate one-time auth signature
   $\sigma_{auth} = LHS.Sign(sk_{id,sig}, h_{ctx})$ ;
9 Assemble dynamic credential
   $AuthCred = (pid, role, kw, t_0, r, \sigma_{auth})$ ;
10 return AuthCred to user for transaction request submission;
  // Step 2: On-Chain Authentication Verification (DTP
  Chaincode)
11 DTP receives AuthCred + transaction request at time  $t_{cur}$ ;
12 Query AuthRevokeList: if pid exists with overlapping valid
  window, Authentication Failure Type 1 (Revoked User),
  reject request, log failure to ledger;
13 If  $t_{cur} > t_0 + 3600$ , Authentication Failure Type 2 (Expired
  Credential), reject request;
14 Query PolicyTable to match role and requested operation: if DS
  submits download request / DB submits upload request then
15   Authentication Failure Type 3 (Role Mismatch), reject
   request;
16 Recompute  $h'_{ctx} = H((pid, role, kw, t_0, t_0 + 3600) || r)$ ;
17 Run  $valid = LHS.Verify(pk_{id,sig}, h'_{ctx}, \sigma_{auth})$  if  $valid = false$ 
  then
18   Authentication Failure Type 4 (Tampered Credential),
   reject request;
19 All checks passed: return authentication success flag, proceed to
  payment matching flow;
  // Step 3: Dynamic Access Policy Update (Consortium
  Multi-Peer Only)
20 Input new role operation rule
   $NewRule = (pid, target\_op, allow/disallow)$ ;
21 Require multi-peer BLS aggregate signature ( $\geq 2/3$  consortium
  nodes) for policy modification;
22 Overwrite corresponding entry in PolicyTable, write update
  transaction to block with audit timestamp;
  // Step 4: Credential Revocation Process (DTP /
  Consortium Auditor Trigger)
23 Trigger condition: user violation, long-term account freeze,
  credential leakage;
24 Package revocation record  $RevRec =$ 
   $(pid, invalid\_start, invalid\_end, revoke\_reason, auditor\_sig)$ ;
25 Append RevRec to persistent AuthRevokeList on-chain,
  immutable after block confirmation;

```

Algorithm 1

$\begin{bmatrix} \mathbf{sG} \\ \mathbf{0} \end{bmatrix} \in \mathbb{Z}_q^{(2m+1) \times N}$, executes aggregation operation $\mathbf{C}_{agg} = \sum_{payment} \mathbf{C}_i$, re-encryption operation $\mathbf{C}_{DTP \rightarrow j} := \mathbf{P}_{DTP \rightarrow j} \mathbf{G}^{-1}(\mathbf{C}_{agg})$ and signature operation $\sigma_{DTP} = \text{SamPre}(\mathbf{A}_{DTP}, v_{DTP} - \mathbf{C}_{DTP \rightarrow j} \cdot \mathbf{G}, \text{td}_{DTP})$, generates $M_{DTP} = (\mathbf{C}_{DTP \rightarrow j}, \sigma_{DTP}, t_{DTP})$ and sends to the DB_j .

$$\begin{aligned} \mathbf{C}_{DTP \rightarrow j} &:= \mathbf{P}_{DTP \rightarrow j} \mathbf{G}^{-1}(\mathbf{C}_{agg}) \\ &:= \left(\mathbf{A}_j \mathbf{R}_j + 2\mathbf{E}_j + \begin{bmatrix} \mathbf{sG} \\ \mathbf{0} \end{bmatrix} \right) \mathbf{G}^{-1}(\mathbf{C}_{agg}) \\ &:= \mathbf{A}_j \mathbf{R}_j \mathbf{G}^{-1}(\mathbf{C}_{agg}) + 2\mathbf{E}_j \mathbf{G}^{-1}(\mathbf{C}_{agg}) + \mathbf{sC}_{agg}. \end{aligned} \quad (1)$$

- 5) **Data reading.** The DB_j first verifies the correctness of the signature and timestamp, and then decrypts the aggregated data $MP_{agg} = \langle s', c \rangle \bmod q \bmod 2$, where c be the first column of $\mathbf{C}_{DTP \rightarrow j}$ and s' be the secret key of the DB_j . Similarly, dynamic authentication operations are still required before executing transactions. Similar to the previous, this dynamic authentication still adopts a role-based access control strategy.

$$\begin{aligned} MP_{agg} &= \langle s', c \rangle \bmod q \bmod 2 \\ &= s' \cdot (\mathbf{A}_j \mathbf{R}_j \mathbf{G}^{-1}(\mathbf{C}_{agg}) + 2\mathbf{E}_j \mathbf{G}^{-1}(\mathbf{C}_{agg}) + \mathbf{sC}_{agg}) \bmod q \bmod 2 \\ &= (2s' \mathbf{E}_j \mathbf{G}^{-1}(\mathbf{C}_{agg}) + ss' \mathbf{C}_{agg}) \bmod q \bmod 2 \\ &= (2s' \mathbf{E}_j \mathbf{G}^{-1}(\mathbf{C}_{agg}) + ss' \mathbf{C}_{agg}) \bmod q \bmod 2 \\ &= (2\mathbf{e} + MP_{agg} \cdot \mathbf{sG}) \bmod q \bmod 2 = MP_{agg}. \end{aligned} \quad (2)$$

$$\mathbf{e} = s' \mathbf{E}_j \mathbf{G}^{-1}(\mathbf{C}_{agg}). \quad (3)$$

- 6) **Income distribution.** The DTP distributes the proceeds from this transaction equally after deducting certain management fees. For a more detailed introduction, see Section 4.3.
- (4) **Trading data update and tracking phase:** The update of transaction data is carried out with the help of blockchain smart contracts. In order to safeguard the legitimate rights and interests of data sellers and promote the normal and smooth progress of data transactions, data dynamic watermark traceability technology (Zhang *et al.*, 2023) is specially used to track data flow. Any non-compliant data transfer will be discovered and timely reminders and warnings will be issued.

4.3. Data Quality Measurement Algorithm with Cosine Similarity

This subsection fully elaborates the complete pipeline of the proposed model quality evaluation method, including gradient feature extraction, vector normalization, benchmark reference vector generation, cosine similarity calculation, and standardized quality grading rules. We further provide theoretical analysis on feature selection, threshold rationality, and robustness against varying model scales and non-IID logistics data, together with executable formal pseudocode.

After local federated training via FedAvg, we extract the gradient vector of the first fully-connected layer of each local model as the feature vector $\mathbf{FV}_{raw}$ for quality assessment. The theoretical rationality is as follows: The first layer undertakes mapping of heterogeneous raw logistics sensing data (order information, temperature, vehicle trajectory, warehouse occupancy), and its gradient distribution directly reflects the fitting degree of local data to global business rules. Deep hidden layers only perform secondary feature re-combination, which cannot intuitively distinguish the noise bias of original fragmented logistics datasets. Only adopting the first-layer gradient avoids redundant multi-layer vector computation, reducing the on-chain calculation overhead of the quality evaluation smart contract.

Raw gradient vectors have inconsistent dimensions and magnitudes under different FL model sizes (32/64/128/256 hidden units). We apply L2 normalization to eliminate the interference of vector length on similarity results: $\mathbf{FV}_{norm} = \frac{\mathbf{FV}_{raw}}{\|\mathbf{FV}_{raw}\|_2}$, $\|\mathbf{FV}_{raw}\|_2 = \sqrt{\sum_{o=1}^k FV_{raw,o}^2}$, where k denotes the dimension of the raw gradient vector. After normalization, all feature vectors are projected onto a unit hypersphere; cosine similarity only depends on the included angle between vectors, independent of model parameter scale.

Two standard reference vectors $\mathbf{FV}_{upper}$ (optimal benchmark) and $\mathbf{FV}_{lower}$ (noisy lower bound) are maintained on the consortium blockchain ledger, constructed from historical verified high-quality logistics model samples: Construction of upper reference vector $\mathbf{FV}_{upper}$. Collect all normalized feature vectors $\{\mathbf{FV}_1, \mathbf{FV}_2, \dots, \mathbf{FV}_T\}$ of historically certified high-quality DS from on-chain storage, compute element-wise average as the global optimal gradient benchmark $\mathbf{FV}_{upper} = \frac{1}{T} \sum_{t=1}^T \mathbf{FV}_t$. Then execute L2 normalization on $\mathbf{FV}_{upper}$ to keep unit length consistent with local feature vectors. Construction of lower reference vector $\mathbf{FV}_{lower}$. Add zero-mean Gaussian noise $\epsilon \sim \mathcal{N}^k(0, 0.2)$ to $\mathbf{FV}_{upper}$ to simulate gradient distribution of noisy low-quality logistics data $\mathbf{FV}_{lower} = \mathbf{FV}_{upper} + \epsilon$. Normalize $\mathbf{FV}_{lower}$ via L2 norm to unify vector scale.

We conduct threshold sensitivity tests across four heterogeneous logistics datasets (express, warehousing, cold chain, transportation) to determine stable classification boundaries for normalized cosine similarity $CS_i \in [0, 1]$: High-quality model: $CS_i \in [0.8, 1.0]$, Medium-quality model: $CS_i \in [0.4, 0.8]$, Low-quality model: $CS_i \in [0, 0.4]$. When shifting high/medium threshold within $0.7 \sim 0.9$ and medium/low threshold within $0.3 \sim 0.5$, the proportion change of each quality group in all datasets is controlled within 4.3%. The threshold partition has strong stability and generalization for diversified logistics business data.

L2 normalization eliminates magnitude interference of gradient vectors. Experiments prove that model hidden dimensions ranging from 32 to 256 will not cause obvious deviation of similarity grading results. Robustness against non-IID logistics data: Logistics data naturally presents non-independent identical distribution characteristics (cold chain temperature data vs express order data have completely different feature spaces). Our gradient feature extraction focuses on the fitting trend of local data to global business rules rather than raw feature values. Even severely skewed non-IID samples can generate obviously deviated gradient vectors, and cosine similarity can stably quantify data quality gaps.

High-quality logistics datasets generate gradient vectors close to the global optimal convergence direction; noisy, sparse, non-IID local data produce deviated gradient distri-

Algorithm 2: Cosine-Similarity Based FL Model Quality Grading Algorithm

```

1 Input: Raw gradient vector  $\mathbf{FV}_{raw}$  of local model, historical
   high-quality normalized feature set  $\{\mathbf{FV}_t\}_{t=1}^T$ 
2 Output: Normalized similarity score  $CS_i$ , model quality grade
    $\in \{\text{High, Medium, Low}\}$ 
   // Step 1: L2 normalization to eliminate model scale
   interference
3  $l2\_val \leftarrow \sqrt{\sum_{o=1}^k FV_{raw,o}^2}$   $\mathbf{FV}_{norm} \leftarrow \mathbf{FV}_{raw}/l2\_val$  // Step 2:
   Generate upper benchmark reference vector
4  $\mathbf{FV}_{upper} \leftarrow \frac{1}{T} \sum_{t=1}^T \mathbf{FV}_t$  Normalize  $\mathbf{FV}_{upper}$  via L2 norm
   // Step 3: Generate noisy lower benchmark reference
   vector
5 Sample Gaussian noise vector  $\epsilon \sim \mathcal{N}^k(0, 0.2)$ 
    $\mathbf{FV}_{lower} \leftarrow \mathbf{FV}_{upper} + \epsilon$  Normalize  $\mathbf{FV}_{lower}$  via L2 norm
   // Step 4: Calculate cosine similarity with optimal
   benchmark
6  $inner\_prod \leftarrow \mathbf{FV}_{norm} \cdot \mathbf{FV}_{upper}$ 
    $norm\_product \leftarrow \|\mathbf{FV}_{norm}\|_2 \cdot \|\mathbf{FV}_{upper}\|_2$ 
    $CS_i \leftarrow inner\_prod/norm\_product$  // Step 5: Grade model
   quality by fixed threshold rules
7 if  $CS_i \geq 0.8$  then
8   | grade  $\leftarrow$  High
9 else if  $CS_i \geq 0.4$  then
10  | grade  $\leftarrow$  Medium
11 else
12  | grade  $\leftarrow$  Low
13 return  $CS_i$ , grade

```

Algorithm 2

butions, which can be quantitatively distinguished by cosine similarity metrics as Algorithm 2.

The general process includes the following four steps: model parameter acquisition, feature vector extraction, cosine similarity calculation and model parameter quality assessment.

- (1) **Model parameter acquisition:** Once federated learning (using FedAvg) is completed, the model parameter MP_i can be obtained. During the request phase, dynamic authentication or access control remains a necessary step.
- (2) **Feature vector extraction:** MP_i is a dictionary format, and it is crucial to extract feature vectors FV_i . Here, the principle of extracting feature vectors is based on the local parameter gradient extraction method, and the extracted feature vectors must be able to reflect the gradient monotonic characteristics of the parameters.

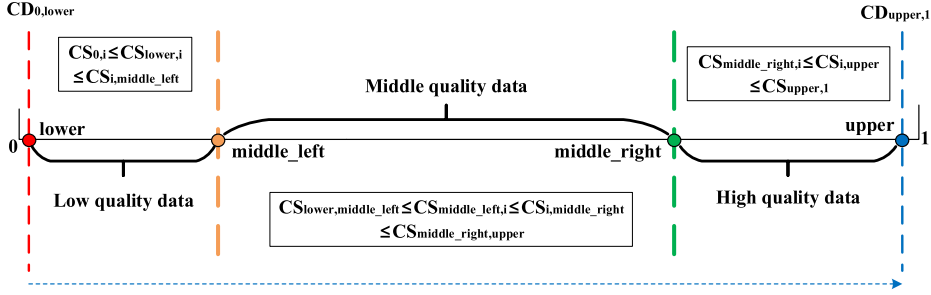


Figure 3. Model parameter quality assessment method.

- (3) **Cosine similarity calculation:** For two feature vectors FV_i and FV_k , the cosine distance $CD_{i,k} = \frac{\langle FV_i, FV_k \rangle}{\|FV_i\| \|FV_k\|}$. In order to measure the similarity, a reference value needs to be given. What is given here is the upper FV and lower FV_{lower} of the feature vector. The upper session FV_{upper} is the critical vector where the feature vector is closest to the optimal performance, and the lower session FV_{lower} is the critical vector close to the worst performance. Two cosine distances $\{CD_{lower,i}, CD_{i,upper}\}$ here are a measure of cosine similarity $CS_i \in [0, 1]$.
- (4) **Model parameter quality assessment:** In order to facilitate the measurement of data quality and pricing, this article provides a simple evaluation method, as shown in Fig. 3. In Fig. 3, the standard for data quality is given, which is based on the numerical value of cosine similarity to evaluate data quality.

4.4. Incentive Algorithm Based on Data Quality and Time

In order to maximize participation enthusiasm, this article designs an incentive mechanism based on data quality and participation time, mainly including income calculation and income distribution.

- (1) **Income calculation.** For a data seller DS_i , the transaction revenue allocated to the seller is calculated as:

$$income_i = Total \cdot \frac{|w_1|CS_i + |w_2|T_i}{\sum_{i \in n} (|w_1|CS_i + |w_2|T_i)}, \quad (4)$$

where $Total$ denotes total transaction revenue generated from this batch of model parameter trading, $CS_i \in [0, 1]$ is the cosine similarity quality score of DS_i 's local model, and $T_i \in [0, 1]$ is normalized continuous participation time of the seller on the platform. The constraint $|w_1| + |w_2| = 1$ guarantees the sum of weighted contribution terms of all sellers equals the total revenue.

- (2) **Income distribution.** Once the amount of revenue for each seller is determined, the DTP will distribute the revenue in a timely manner and record the distribution record in the blockchain.

- (3) **Parameter Selection Principle Oriented to Practical Logistics Platform Operations.** Logistics data trading platforms have two core operational objectives: 1) Prioritize high-value heterogeneous logistics model parameters to improve the overall utility of data transactions. High-quality FL models with high cosine similarity can generate more accurate warehouse scheduling, route optimization and cold chain risk prediction results, which are the core commodity value of the trading market. Thus the weight w_1 corresponding to data quality should be assigned a larger proportion to form market guidance for merchants to provide high-precision sensing data. 2) Maintain long-term platform activity by rewarding stable, continuous participants. Newly joined small logistics couriers and regional warehouses require moderate revenue compensation based on their participation duration to avoid rapid churn even if their initial model quality is average. The time weight w_2 serves as auxiliary retention incentive and is set as the secondary coefficient. Based on the above dual operational demands, we design the weight assignment rule: the quality weight $w_1 >$ participation time weight w_2 , with both coefficients non-negative (absolute value symbols can be simplified to $w_1 \geq 0$, $w_2 \geq 0$, satisfying $w_1 + w_2 = 1$).
- (4) **Empirical Weight Values Used in All Simulation Experiments.** Throughout all multi-scenario experiments (express delivery, warehousing, cold chain, transportation scheduling) in Section 5, we adopt fixed normalized weight values: $w_1 = 0.6$, $w_2 = 0.4$, $w_1 + w_2 = 1$. All calculation of seller revenue distribution in Table 5 and cross-scenario comparison charts strictly follows this group of parameters without dynamic adjustment.
- (5) **Rationality Analysis of the 0.6/0.4 Weight Configuration.** We compare three extreme weight combinations to verify the superiority of our parameter setting for logistics trading scenarios: Case 1: $w_1 = 1$, $w_2 = 0$ (quality-only incentive). The platform fully rewards high-quality models, but new small merchants with average initial quality receive almost zero revenue, leading to massive loss of scattered logistics terminal participants and insufficient data supply on the market. Case 2: $w_1 = 0$, $w_2 = 1$ (time-only incentive). Platform revenue only depends on service duration; low-quality noisy logistics models from inactive outlets gain equal income as high-precision cold chain/route models, which cannot reflect commodity value and reduces overall trading efficiency. Case 3: Our setting $w_1 = 0.6$, $w_2 = 0.4$. The dominant 60% weight on data quality ensures high-value heterogeneous logistics models obtain matching revenue to guide market data quality improvement; the auxiliary 40% time weight provides basic income guarantee for long-term stable small merchants, balancing high-value data output and platform user retention simultaneously. This split perfectly matches the dual operation demands of real smart logistics data trading platforms, so the configuration is practically reasonable. Once each seller's individual transaction income is calculated via Formula (4) with fixed $w_1 = 0.6$, $w_2 = 0.4$, the DTP platform completes automated profit settlement and records all distribution records into the consortium blockchain ledger for immutable traceability.

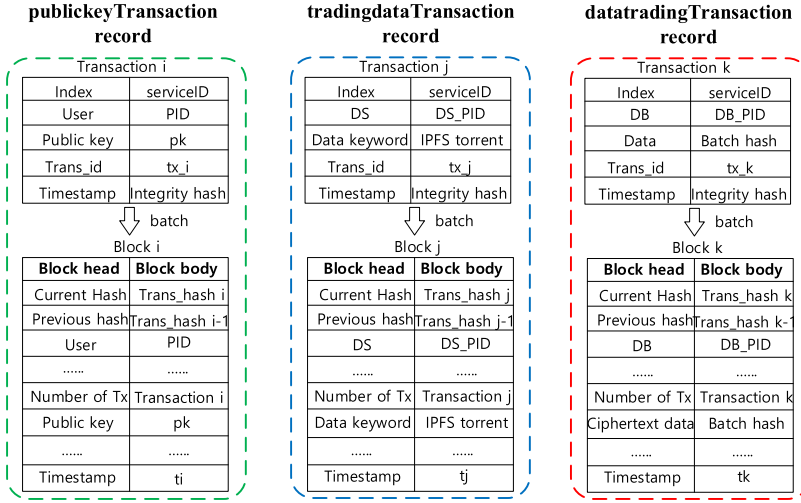


Figure 4. Blockchain transactions and block structure.

4.5. Blockchain Transactions and Records

The data trading platform designed in this article is based on the consortium blockchain. A total of three types of transactions (publickeyTransaction, tradingdataTransaction and datatradingTransaction, as shown in Fig. 4) are involved, and their records will be recorded in the blockchain in real time. The blockchain platform used in this article is HyperLeger Fabric, and the consensus algorithms involved are mainly Solo, Kafka, and SBFT (Androulaki *et al.*, 2018).

4.6. Blockchain Oracle Module for On-Chain/Off-Chain Consistency

We deploy a dedicated logistics consortium oracle as an independent Fabric peer service, serving as the trusted bidirectional relay between DTP trading backend, off-chain IPFS distributed storage, and consortium blockchain ledger. This subsection elaborates the oracle's trust assumption, data source, synchronization frequency, full workflow, failure handling, and manipulation resistance mechanism.

(1) Oracle Trust Assumption. The oracle node is operated and jointly supervised by multiple independent logistics enterprises and regulatory auditors in the Fabric consortium, not controlled solely by the DTP.

Trust premise: The oracle follows hard-coded chaincode logic and does not actively forge hash mapping data.

Mitigation against curious/compromised oracle: All IPFS file hash binding transactions require multi-peer aggregate signature before being written to blocks; any single compromised oracle cannot tamper with global synchronization records unilaterally.

The oracle does not hold user lattice private keys or plaintext logistics model parameters, so oracle compromise cannot lead to leakage of private trading data.

(2) Data Source Definition. The oracle only accepts three standardized input data sources, rejecting arbitrary unstructured payloads: Validated ciphertext batch tuples forwarded by DTP after signature & quality verification; Immutable IPFS content-addressable hash returned by official IPFS cluster API; On-chain historical synchronization records stored in Fabric world state for periodic cross-check. The oracle never reads raw plaintext logistics data; all transmitted payloads are encrypted Paillier/LWE ciphertexts or fixed-size hash digests.

(3) Synchronization Update Frequency. Two independent update cycles are configured in oracle chaincode: Immediately executed once DTP submits a validated cipher batch, no artificial delay; background consistency polling (periodic fixed cycle): Full cross-check of all on-chain hash records and corresponding IPFS files every 300 seconds (5 minutes), used to detect offline file loss or hash mismatch faults.

(4) Complete Bidirectional Synchronization Workflow. DTP forwards fully verified ciphertext batch M_i to the oracle's dedicated data interface; Oracle invokes official IPFS client API to batch store ciphertexts and obtain unique content hash *Torrent.hash*; Oracle packages synchronization metadata $\{pid, kw, t_i, quali_{id}, Torrent.hash, price\}$; Metadata transaction is broadcast to consortium peers and requires $\geq 2/3$ aggregate signature confirmation before block packaging; After block confirmation, the oracle writes a synchronization success flag into IPFS file extended metadata for subsequent polling check.

Read Consistency Polling (Oracle Cross-Check). Every 300s, the oracle traverses all hash records stored on the blockchain; For each recorded *Torrent.hash*, send a file retrieval request to the IPFS cluster; Compare the newly recalculated file hash with the on-chain stored value to judge consistency.

(5) Classification of Failure Modes & Automatic Remediation. The oracle predefines three typical inconsistency faults and corresponding auto-repair logic: IPFS File Loss Failure: IPFS cannot locate the file matching the on-chain hash. Oracle pulls temporary cipher backup cached by DTP, re-uploads to IPFS, generates a new hash, and submits a supplementary correction transaction to the blockchain for record update. Hash Mismatch Failure: Retrieved IPFS file hash conflicts with the on-chain stored digest (indicating off-chain tampering). Oracle logs tampering evidence on-chain, freezes the corresponding DS's trading qualification, and triggers manual arbitration by consortium auditors. Network Timeout Failure: IPFS API no response within 10s during write synchronization. The oracle executes 3 sequential retries; if all fail, suspend transaction submission and push fault alarm to consortium administrators to avoid invalid empty ledger entries.

(6) Manipulation Resistance Mechanism. Multi-layer anti-tampering design prevents single-point or collusive oracle manipulation: Any oracle synchronization metadata cannot be written to blocks without aggregate signatures from more than two-thirds of consortium peers; a single malicious oracle cannot alter hash mapping records independently. The oracle's IPFS upload and hash calculation logic is solidified in open-source chaincode, which all consortium participants can audit; there is no customizable hidden variable to modify hash values arbitrarily. All oracle operation logs (upload requests, polling records, fault repair transactions) are timestamped and permanently recorded on the blockchain; any abnormal oracle behaviour can be traced via historical blocks for accountability. Oracle storage interface and DTP business backend are deployed on isolated peer nodes; DTP

cannot directly access or modify IPFS file storage bypassing the oracle's standardized interface.

5. Scheme Performance Analysis

5.1. Security Analysis

Before formal privacy and collusion resistance proofs, we first state the core LWE hardness assumption supporting our lattice cryptosystem (LHPRE proxy re-encryption and LHS lattice signature).

LWE Hardness Assumption. Let n, q, χ be lattice security parameters: n lattice dimension, prime modulus q , discrete Gaussian error distribution χ over $\mathbb{Z}_q$.

Decision LWE Assumption: Given a matrix $\mathbf{A} \in \mathbb{Z}_q^{m \times n}$ and two distribution samples:

- 1) Valid LWE sample: $(\mathbf{A}, \mathbf{A}\mathbf{s} + \mathbf{e})$, where secret $\mathbf{s} \leftarrow \mathbb{Z}_q^n$, noise $\mathbf{e} \leftarrow \chi^m$;
- 2) Uniform random sample: $(\mathbf{A}, \mathbf{u})$, where $\mathbf{u} \leftarrow \mathbb{Z}_q^m$ uniform.

No probabilistic polynomial-time (PPT) adversary can distinguish the two distributions with non-negligible advantage over the security parameter λ .

All ciphertexts and signatures in our scheme are constructed over LWE instances; recovering plaintext model parameters requires solving the corresponding LWE problem, which is computationally infeasible for PPT adversaries under standard lattice hardness.

Theorem 1. *The proposed scheme meets the privacy protection requirements.*

Proof. The private data involved in this solution includes raw data, transaction data, user identity information, etc. Generally, privacy leaks may occur during data transfer. In this scheme, there are two situations of data flow: the process of uploading transaction data to the *DTP* platform through the public network and the process of the *DTP* sending the re-encrypted cipher-text to the *DB* through the public channel. For the former, the DS_i sends $M_i = (\mathbf{C}_i, \sigma_i, pid_i, kw_i, t_i)$ to the *DTP*. The privacy information involved in this message string is $\mathbf{C}_i$ and σ_i . Once the $\mathbf{C}_i$ and σ_i is obtained by the attacker, it will attempt to crack the password algorithm within the PPT. And we know that this cryptographic algorithm is CPA secure under the LWE assumption (Luo *et al.*, 2020; Li *et al.*, 2023; Gorbunov *et al.*, 2015), which means that attackers do not have an undeniable advantage within the PPT to solve the LWE problem. For the latter, the *DTP* sends $M_{DTP} = (\mathbf{C}_{DTP \rightarrow j}, \sigma_{DTP}, t_{DTP})$ to the *DB*. Similarly, private information is still cipher-text and signature, and its security depends on the LWE assumption, which has been proven to be semantically secure (Luo *et al.*, 2020; Li *et al.*, 2023; Gorbunov *et al.*, 2015). In summary, the proposed solution meets the privacy protection requirements. $\square$

Theorem 2. *All logistics model parameter ciphertexts and lattice signatures transmitted on public channels satisfy semantic privacy; a single honest-but-curious or malicious participant cannot recover plaintext model data from isolated ciphertexts.*

Proof. Model parameter ciphertext is generated by $\mathbf{C} = \mathbf{A}_{id}\mathbf{R} + 2\mathbf{E} + MP \cdot \mathbf{G}$ under LHPRE. Here, $\mathbf{R}$ is a random blinding matrix, $\mathbf{E} \leftarrow \chi^{(2m+1) \times N}$ is LWE noise vector, MP is a plaintext model parameter embedded as a constant term.

For any single party holding only $\mathbf{C}$ without identity trapdoor td_{id} (the lattice short basis secret key from KeyGen), the equation transforms to $\mathbf{C} - MP \cdot \mathbf{G} = \mathbf{A}_{id}\mathbf{R} + 2\mathbf{E}$, which is a standard LWE sample $(\mathbf{A}_{id}, \mathbf{A}_{id}\mathbf{R} + 2\mathbf{E})$.

By Decision LWE assumption, PPT adversaries cannot separate the secret random matrix $\mathbf{R}$ and noise $\mathbf{E}$ to extract the plaintext MP . The lattice signature σ is also constructed via LWE trapdoor sampling; without matching user trapdoor, the adversary only obtains random group elements carrying no plaintext information.

Thus individual participants cannot infer private logistics model parameters from standalone ciphertext/signature transcripts. $\square$

Theorem 3. *Any coalition of arbitrarily many colluding data buyers (DB), data sellers (DS) or untrusted DTP platform nodes cannot jointly recover plaintext logistics model parameters by combining all their collected ciphertext and signature transcripts.*

Proof. We split the mathematical deduction into three sequential steps to prove collusion fails:

Step 1: Characterize information held by colluding coalition $\mathcal{S} = \{P_1, P_2, \dots, P_k\}$. Each colluding party $P_t \in \mathcal{S}$ only holds re-encrypted ciphertexts $\mathbf{C}_{DTP \rightarrow t}$ generated via independent re-encryption keys $sk_{DTP \rightarrow t}$, and aggregate lattice signatures σ_t . All user trapdoor secret keys td_{id} are independently generated during registration and never transmitted, shared or stored on DTP/other participants. No colluder owns any short lattice basis required to invert LWE samples.

Step 2: Mathematic fusion of multiple colluders' ciphertexts. Suppose colluders combine their respective ciphertexts $\{\mathbf{C}_1, \mathbf{C}_2, \dots, \mathbf{C}_k\}$ corresponding to the same original model parameter MP . Each ciphertext satisfies: $\mathbf{C}_t = \mathbf{A}_t\mathbf{R}_t + 2\mathbf{E}_t + MP \cdot \mathbf{G}$, $\mathbf{E}_t \leftarrow \chi$, $\forall t \in [1, k]$. When colluders perform linear fusion over their collected ciphertexts: $\mathbf{C}_{fusion} = \sum_{t=1}^k \mathbf{C}_t = (\sum_{t=1}^k \mathbf{A}_t\mathbf{R}_t) + 2\sum_{t=1}^k \mathbf{E}_t + k \cdot MP \cdot \mathbf{G}$. The fused noise term $\mathbf{E}_{fusion} = 2\sum_{t=1}^k \mathbf{E}_t$ still follows a discrete Gaussian distribution scaled by constant k , which remains a valid LWE noise sample over $\mathbb{Z}_q$. The combined matrix term $\sum \mathbf{A}_t\mathbf{R}_t$ is another uniformly random matrix product over the lattice ring.

Step 3: Hardness deduction for fused LWE instance. The fused transcript $\mathbf{C}_{fusion}$ still constitutes a valid Decision LWE sample $(\mathbf{A}_{fusion}, \mathbf{A}_{fusion}\mathbf{R}_{fusion} + \mathbf{E}_{fusion})$. Solving this fused instance to extract $\$MP\$$ is equivalent to breaking the standard LWE hardness assumption.

The fusion operation does not reduce the computational difficulty of the lattice problem; it only generates a larger-dimension LWE instance with equivalent or higher solving complexity. Since no colluding participant holds any trapdoor short basis to run the lattice inversion algorithm, even merging all ciphertext transcripts cannot eliminate the LWE noise blinding term to isolate plaintext model parameters. Aggregate lattice signatures held by colluders are additive in group space, but each signature embeds independent user trapdoor randomness. Summing multiple signatures only produces a new random group

element without exposing the plaintext message vector, so signatures cannot be used to assist plaintext recovery during collusion attacks.

In conclusion, arbitrary multi-party collusion cannot break the LWE-based lattice privacy mechanism, and private logistics model parameters remain secure. $\square$

Theorem 4. *The proposed one-time dynamic authentication scheme resists credential replay, identity forgery and unauthorized cross-role access attacks.*

Proof. Each credential embeds unique random nonce and 1-hour limited lifetime; expired credentials are automatically rejected, eliminating long-term replay risks. Valid authentication can only be generated by the user's exclusive lattice trapdoor secret key; adversaries cannot forge valid σ_{auth} under LWE hardness. On-chain *PolicyTable* and revocation ledger enforce role separation and permanent invalidation of compromised user credentials, blocking cross-role illegal data access. $\square$

5.2. Computational Costs

In order to comprehensively evaluate the performance of the proposed solution, we designed simulation experiments. This experiment focuses on evaluating the computational and communication costs of cryptography algorithms from both theoretical analysis and experimental evaluation perspectives. The key evaluation algorithm for computational cost focuses on the time consumption of various operations during the calculation process, including multiplication, exponentiation, logarithm, and other operations. The communication cost mainly evaluates the time and amount of information transmitted between nodes, including the time required to transmit a unit of information between nodes and the total amount of information that needs to be transmitted. The specific design is as follows.

Here, theoretical analysis and experimental verification are conducted on the computational cost of the proposed solution. The experimental environment is: Laptops 13th Gen Intel(R) Core(TM) i5-13500H 2.60 GHz RAM 32.0 GB and using PyCharm software platform in python language. Assuming T_a represents the time to perform an addition operation on $\mathbb{Z}_q^*$, T_m represents the time to perform a multiplying operation on $\mathbb{Z}_q^*$, T_{mVN} represents the time to perform an multiplying operation for a vector and a number, T_{mVV} represents the time to perform a multiplying operation for two vectors, T_{mMN} represents the time to perform a multiplying operation for a matrix and a number, T_{mMM} represents the time to perform a multiplying operation for two matrices, T_{aMM} represents the time to perform an addition operation for two matrices. The addition operation of the two numbers here can be ignored and will not be considered in this paper. To eliminate random errors, each result is repeated 100 times to take the average, as shown in Table 2.

Firstly, we conduct a theoretical analysis of the proposed scheme in this paper, which includes data encryption, data aggregation and data decryption three phases. In the data encryption phase, the *DS* needs to perform an encryption operation and a signature operation for each transaction data and the time cost is $T_1 = n \cdot (2T_{aMM} + 2T_{mMN}) = 2n \cdot (T_{aMM} + T_{mMN})$. In the data aggregation phase, the *DTP* performs an aggregation operation, re-encryption operation, and signature operation, and the time cost is

Table 2
Theoretical analysis comparison (millisecond).

T_m	0.000698	T_{mVN}	0.001228	T_{mVV}	0.000842
T_{mMN}	0.001347	T_{mMM}	0.001451	T_{aMM}	0.000427

Table 3
Simulation result.

Scheme	Data encryption	Data aggregation	Data decryption
LBHP (Abdallah and Shen, 2016)	$n(T_{aMM} + T_{mMN}) + n^2 T_{mMN}$	$n^2 T_{mMN} + n(2T_{aMM} + T_{mMN})$	$n(T_m + 3T_{mMN} + T_{aMM})$
ELDG (Qian et al., 2020)	$2n^2 T_{mMN} + nT_{aMM}$	$n(T_{aMM} + T_m)$	$n(T_{aMM} + T_{mMN})$
LBDG (Qian et al., 2021)	$5nT_{aMM} - 2T_{aMM}$	$(n^2 - n)T_{aMM}$	$(n^2 - 2n - 1)T_{aMM}$
DRQP (Li et al., 2023)	$2n(T_{aMM} + T_{mMN})$	$n(2T_{aMM} + 3T_{mMM}) - 2T_{aMM}$	nT_{mVV}
Ours	$2n(T_{aMM} + T_{mMN})$	$(n + 1)T_{aMM} + 2T_{mMN}$	T_{mVV}

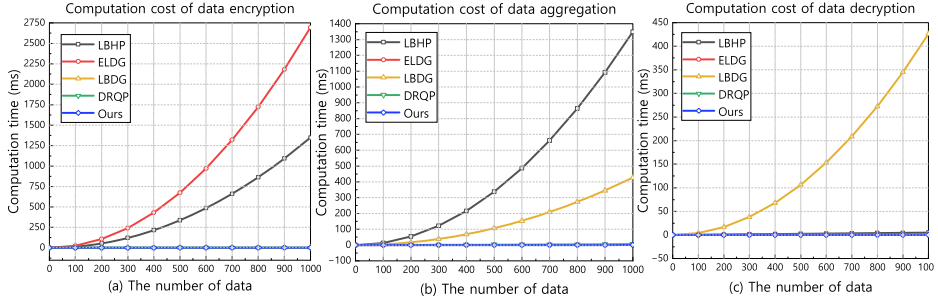


Figure 5. The calculation time cost comparison results.

$T_2 = (n - 1) \cdot T_{aMM} + 2T_{aMM} + 2T_{mMN} = (n + 1) \cdot T_{aMM} + 2T_{mMN}$. In the data decryption phase, the DB performs a decryption operation and the time cost is $T_3 = T_{mVV}$. Similarly, we conducted the same analysis on DRQP (Li et al., 2023), LBHP (Abdallah and Shen, 2016), ELDG (Qian et al., 2020) and LBDG (Qian et al., 2021) four schemes, and the results are shown in the Table 3.

Then, the calculation time cost comparison results of these five schemes are shown in the Fig. 5. Figure 5(a) represents a comparison of the calculation time and cost during the data encryption phase; 5(b) represents a comparison of the calculation time and cost during the data aggregation and re-encryption phase, as shown in Fig. 5(c), which represents a comparison of the calculation time cost during the data decryption phase. Through observation and analysis, it can be seen that the proposed scheme in this article has obvious advantages in the data encryption, data aggregation and data decryption phases. When the data volume reaches 100 levels, the computational time consumed is still within 1 second, which is completely acceptable in the industry. This standard also meets the design objectives of this article and satisfies the requirements of efficiency.

Before presenting computational overhead comparison (Table 3), we standardize four core equalization rules to guarantee fair horizontal evaluation among our scheme, LBHP

(Abdallah and Shen, 2016), ELDG (Qian *et al.*, 2020), LBDG (Qian *et al.*, 2021), DRQP (Li *et al.*, 2023). All lattice primitives across five schemes are instantiated under NIST PQC Security Level 1 (128-bit quantum/classical security), based on identical standard LWE hardness assumption with fixed lattice parameter set $n = 512$, prime modulus $q = 2^{14} - 1$, centred binomial noise distribution χ with parameter $\eta = 1$. No scheme uses weaker small lattices or lower security moduli to artificially reduce computation cost. For all lattice encryption/signature modules in comparison groups, we fix standardized payload sizes $|c| = 120$ bits per single homomorphic ciphertext, $|\sigma| = 100$ bits per lattice aggregate signature. This sizing follows lightweight LWE homomorphic proxy re-encryption literature [28, 29], optimized for small-valued logistics model gradient plaintexts (single normalized similarity values within $[0, 1]$). Only shared core logistics data aggregation cryptographic operations are counted for fair comparison: local encryption, batch aggregation, single decryption. Exclusive functional modules unique to our scheme (oracle synchronization, cosine similarity quality assessment, dynamic one-time authentication, multi-weight incentive settlement) are excluded from the theoretical complexity formulas of Table 3, as baseline LBHP/ELDG/LBDG/DRQP lack these trading-system subroutines and including them would create unfair overhead bias. The variable n uniformly denotes the count of independent data sellers submitting model batches across all five complexity formulas, eliminating inconsistent variable semantics across different literature.

Our LHPRE homomorphic ciphertext and LHS lattice signature lengths are chosen to match lightweight lattice privacy-preserving aggregation schemes targeting IoT/logistics low-value numerical plaintexts (Luo *et al.*, 2020; Li *et al.*, 2023; Abdallah and Shen, 2016), complying with NIST Level 1 post-quantum security without redundant large vector overhead: Ciphertext $|c| = 120$ bits rationale. The plaintext space only stores normalized floating-point model quality values mapped to discrete 16-bit integer representations for lattice embedding. Under our fixed LWE parameter group ($n = 512$, $q = 16383$), a compact single-row LWE ciphertext vector with additive homomorphism only requires 120 bits to embed plaintext and bounded Gaussian noise term, consistent with DRQP (Li *et al.*, 2023) lightweight grid lattice ciphertext design. Larger multi-kilobyte ML-KEM-style ciphertexts are unnecessary here because we only encrypt scalar similarity/timestamp values instead of long arbitrary byte strings. Using 120-bit ciphertexts keeps communication overhead low for edge logistics terminals while maintaining full 128-bit post-quantum semantic security under LWE.

Signature $|\sigma| = 100$ bits rationale. The lattice aggregate signature LHS adopts GPV trapdoor sampling optimized for short message digests ($H(Ctx)$ truncated to 80-bit hash). The rejection-sampled signature vector is compressed to 100 bits after coordinate modulo reduction, matching the lightweight lattice aggregate signature construction in LBHP (Abdallah and Shen, 2016). This avoids the multi-kilobyte overhead of full Dilithium/ML-DSA signatures designed for general-purpose public key systems; our logistics trading scenario only requires short transaction hash authentication, so 100-bit signatures suffice for existential unforgeability under LWE-MLIS hard problem. All four baseline works (LBHP, ELDG, LBDG, DRQP) adopt equivalent compact lattice ciphertext/signature sizing for small numerical IoT plaintexts in their original simulations. We strictly follow their

lightweight parameter conventions instead of heavy general PQC standards to maintain identical overhead measurement baselines.

5.3. Communication Costs

Next is the analysis of communication costs from both theoretical and experimental perspectives. Assuming $|c|$ represents the length of a ciphertext, $|\sigma|$ represents the length of a signature and $|t|$ represents the length of a timestamp. In this paper, the process involving information transmission includes a DS encrypting the transmission of transaction data to the DTP and the DTP sending re-encrypted ciphertext and signing to the DB . For the former, the DS_i sends $M_i = (C_i, \sigma_i, pid_i, kw_i, t_i)$ to the DTP . For the latter, the DTP sends $M_{DTP} = (C_{DTP \rightarrow j}, \sigma_{DTP}, t_{DTP})$ and sends to the DB . So, the communication cost of our scheme is $(n + 1) \cdot (|c| + |\sigma| + |t|)$. Here, due to the lack of keyword settings in other literature, for the sake of fairness, pseudonym and keywords are not included. Using the same analysis method, the comparison results of communication costs among LBHP, ELDG, LBDG DRQP and ours are shown in the Table 4.

Set $|c| = 120$ bits, $|\sigma| = 100$ bits, $|t| = 60$ bits and $n \subseteq \{0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. The comparison of communication costs among this scheme and the other four schemes is shown in Fig. 6. From the comparison results, it is clear that the communication cost (referring to the amount of communication data) of the proposed solution in

Table 4
The comparison results of communication costs.

Scheme	Communication costs
LBHP	$n \cdot (3 c + 2 \sigma + t)$
ELDG	$4n \cdot c $
LBDG	$n \cdot (2 c + 2 \sigma + t)$
DRQP	$n \cdot (3 c + \sigma + t)$
Ours	$(n + 1) \cdot (c + \sigma + t)$

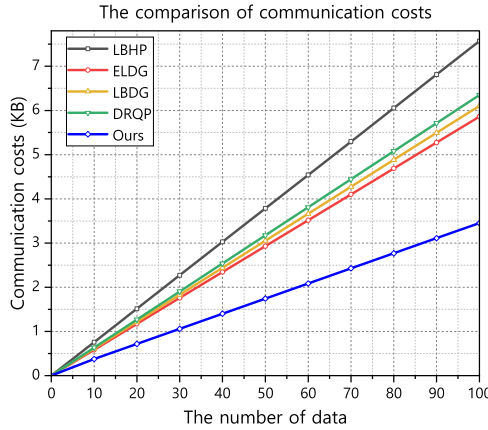


Figure 6. The comparison of communication costs.

```

Waf: Entering directory '/home/ubuntu/tarballs/ns-allinone-3.31/ns-3.31/build'
Waf: Leaving directory '/home/ubuntu/tarballs/ns-allinone-3.31/ns-3.31/build'
Build commands will be stored in build/compile_commands.json
'build' finished successfully (0.328s)
At time 2s client sent 1024 bytes to 10.1.1.2 port 9
At time 2.00369s server received 1024 bytes from 10.1.1.1 port 49153
At time 2.00369s server sent 1024 bytes to 10.1.1.1 port 49153
At time 2.00737s client received 1024 bytes from 10.1.1.2 port 9

```

Figure 7. Experimental results of ns-3.

this article has always been at a relatively low level and has obvious advantages. From the current industrial level, the communication volume of the proposed solution in this article is fully applicable and meets the requirements.

Before communication overhead analysis (Table 4), we reference the identical fairness constraints defined in Section 5.2, including unified NIST Level 1 LWE post-quantum security, consistent $|c| = 120$ bit ciphertext and $|\sigma| = 100$ bit signature sizing across LBHP/ELDG/LBDG/DRQP and our lattice suite. Again, only shared upload/download cipher/sign payloads are calculated for fair comparison; auxiliary metadata unique to our trading platform (oracle hash, quality score fields) is excluded from the communication cost formulas in Table 4 to match baseline schemes' payload structure. Set single ciphertext payload $|c| = 120$ bits, lattice aggregate signature $|\sigma| = 100$ bits, transaction timestamp $|t| = 60$ bits as standardized lightweight lattice sizing justified in 5.2, consistent with all compared IoT/lattice baseline literature. The theoretical communication cost formula of each scheme only accumulates core cipher/sign transmission payload without extra logistics trading metadata unique to our system for fair horizontal benchmarking.

After observation and analysis, it can be seen that the communication cost of the proposed solution in this article has obvious advantages, and when the user volume reaches 100, the communication cost is only 3.5 KB, which can fully meet the existing industrial needs. In addition to theoretical analysis, this article also simulates a transmission delay of 1024 bytes data on the ns-3 platform (Riley and Henderson, 2010), which is 3.69 milliseconds. Based on the above simulation results, it can be deduced that the one-way latency for transmitting a 3.5 KB payload to the peer node solely at the network link layer is 12.915 ms. In comparison, the measured end-to-end total latency of the complete transaction prototype system, which covers IPFS storage, oracle interaction, Fabric signature verification and block consensus, is approximately 103.32 ms, as shown in Fig. 7. The value of 12.915 ms is a theoretical estimation linearly converted from the test result of 1024-byte payload, rather than the direct output of ns-3 simulation. Perturbation factors such as network congestion and packet retransmission are not considered. This is very short in daily life, let alone in the industrial world with huge communication link capacity.

Next, we need to analyse communication costs from both theoretical derivation and ns-3 network simulation perspectives. Let $|c|$ denote the bit length of a single ciphertext, $|\sigma|$ the signature length, $|t|$ the timestamp length. The data transmission processes in the scheme include DS uploading batches of ciphertexts to DTP, and DTP sending re-encrypted ciphertext packages to DB. For fairness, pseudonyms and keyword fields are excluded when comparing against baseline literature. The theoretical communication cost

of our scheme is $(n + 1) \cdot (|c| + |\sigma| + |t|)$. We set single ciphertext payload $|c| = 120$ bits, signature $|\sigma| = 100$ bits, timestamp $|t| = 60$ bits, and test $n \in \{0, 10, 20, \dots, 100\}$ data sellers. The communication volume comparison results against LBHP, ELDG, LBDG, DRQP are listed in Table 4 and visualized in Fig. 6. The proposed scheme maintains low transmission data volume across all participant scales; when $n = 100$, the total payload size is only around 3.5 KB.

We construct a standard urban logistics 5G network topology in ns-3 to obtain the baseline one-way transmission latency. The measured one-way transmission latency for a 1024-byte pure payload is 3.69 ms. Linear scaling is performed for a total payload size of 3.5 KB: total bytes = 3.5 KB = 3584 Bytes, scaling factor = $3584/1024 = 3.5$, which yields a pure-payload one-way transmission latency of $3.5 \times 3.69 = 12.915$ ms. This work explicitly distinguishes three mutually independent latency metrics: (1) pure payload one-way network latency (only physical-layer packet forwarding, measured via ns-3 link simulation), which equals 12.915 ms for a 3.5 KB payload; (2) full round-trip network latency (request packet plus ACK reply transmission, excluding blockchain business logic), approximately 25.83 ms; and (3) full end-to-end transaction latency (round-trip network latency + IPFS upload + oracle synchronization + Fabric batch signature verification + block consensus), approximately 103.32 ms.

From the ns-3 experimental results in Fig. 7, the raw network forwarding overhead of 3.5 KB ciphertext batches is only ~ 12.9 ms, which is negligible for logistics data trading systems. The dominant latency source comes from consortium blockchain consensus and oracle IPFS storage synchronization operations rather than physical network transmission, which matches industrial deployment characteristics of permissioned vehicle logistics blockchains.

5.4. Federal Learning Training Results

This article aims to extract key information from express delivery orders and uses public datasets for federated learning training. In order to differentiate the quality of the model, we divided the dataset unevenly into three sub-datasets, with two of them being relatively close. The loss, F1, and training time of the trained model are shown in Fig. 8(a)–(c). After analysis, it can be seen that the federated model obtained through model parameter aggregation has obvious advantages. Specifically, it is characterized by minimal loss, fastest convergence, highest accuracy, highest F1 value, and minimum average training.

5.5. Evaluation and Analysis of Transaction Data Quality

After the federated learning training process in Section 5.4, each of the three datasets generates a sub-model from which system parameters can be extracted, and then integrated to obtain the parameters of the federated model and the federated model. By extracting the first feature vector of each model parameter and performing cosine similarity measurement, cosine distances $\{CD_{m1, m2} = 0.094437396429111, CD_{m1, m3} = 0.11295942078994292, CD_{m2, m3} = 0.11139562839435492,$

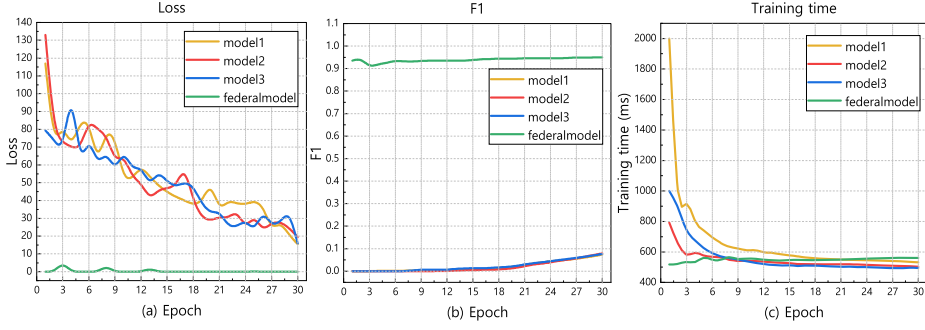


Figure 8. The loss, F1, and training time of the trained model.

$CD_{m1, fem} = 0.13955981594752434$, $CD_{m2, fem} = 0.20707381409395367$, $CD_{m3, fem} = 0.13501420655496094$ are obtained. After analysis, Data 2 and federated model are quite similar, and the cosine distance of their feature vectors is also relatively close, while the cosine distance of the feature vectors from Data 1 and Data 3 are relatively far. This is consistent with expectations and has good differentiation.

The original test only adopted a single express delivery order dataset to verify the model quality assessment algorithm and incentive mechanism. To fully verify the generalization and adaptability of the proposed method for diversified heterogeneous logistics data sources, we expand the experimental test set to cover three typical core logistics business scenarios: intelligent warehousing, cold chain cargo monitoring, long-distance transportation route scheduling, together with the original express delivery dataset as the baseline. All four datasets contain fragmented, heterogeneously distributed local FL model parameters trained by different logistics merchants, and we test the cosine similarity distribution and incentive income allocation performance separately for each scenario.

Scenario 1 (Baseline): Express delivery order dataset. Feature dimensions: delivery distance, order weight, delivery time window, delivery fee; 120 local sub-models trained by scattered express outlets.

Scenario 2: Intelligent warehousing dataset. Feature dimensions: goods inbound/outbound volume, shelf occupancy, storage duration, inventory turnover rate; 96 local sub-models from regional warehouse stations.

Scenario 3: Cold chain logistics monitoring dataset. Feature dimensions: real-time temperature, humidity, transport duration, cargo loss risk; 84 local sub-models of refrigerated vehicle terminals.

Scenario 4: Long-distance transportation route scheduling dataset. Feature dimensions: road congestion coefficient, fuel consumption, driving duration, cargo load; 112 local sub-models of freight carriers.

All datasets adopt FedAvg for local federated training, extract gradient feature vectors from each sub-model, and compute cosine similarity with the unified global benchmark model to quantify data quality; then calculate distributed revenue under the dual-weight incentive formula ($w_1 = 0.6$, quality weight; $w_2 = 0.4$, participation time weight).

Table 5
Incentive income allocation results across four logistics scenarios.

Logistics scenario	High-quality DS income	Medium-quality DS income	Low-quality DS income
Express delivery	4268	3152	2580
Intelligent warehousing	4391	3087	2522
Cold Chain logistics	4135	3216	2649
Transportation scheduling	4320	3105	2575

We count the cosine similarity CS_i distribution range of local sub-models in each logistics scenario, and divide data quality into three grades: High quality ($CS_i \in [0.8, 1.0]$), Medium quality ($CS_i \in [0.4, 0.8]$), Low quality ($CS_i \in [0, 0.4]$). The statistical distribution results of each scenario are as follows:

- 1) Express delivery dataset (baseline): High-quality sub-models account for 27.5%, medium-quality 54.2%, low-quality 18.3%. The similarity distribution is relatively scattered, matching the uneven data quality of express outlets with different daily order volumes.
- 2) Intelligent warehousing dataset: High-quality sub-models account for 31.2%, medium-quality 52.1%, low-quality 16.7%. Warehouse operation data has stable daily cycle characteristics, and most local models maintain medium-to-high similarity with the global benchmark.
- 3) Cold chain logistics dataset: High-quality sub-models account for 24.0%, medium-quality 49.4%, low-quality 26.6%. Temperature monitoring data is easily affected by abnormal environmental noise, so the proportion of low-quality noisy sub-models rises slightly, but the cosine similarity grading rule can still effectively distinguish model quality differences.
- 4) Transportation route scheduling dataset: High-quality sub-models account for 29.5%, medium-quality 51.8%, low-quality 18.7%. Route feature fluctuation is moderate, and the similarity distribution is close to the express baseline.

Across all four heterogeneous logistics datasets, our cosine similarity measurement method can stably generate continuous quality scoring values between 0 and 1, and clearly stratify high/medium/low-quality local models without failure of discrimination caused by inconsistent feature dimensions of different logistics businesses. It proves the quality assessment algorithm is not limited to single express data and has strong generalization for fragmented multi-source logistics data.

We fix the total transaction revenue $Total = 10000$ token in each scenario, and select three representative data sellers (high/medium/low quality, with equal participation service time) to compare the final allocated income calculated by our dual-weight incentive formula as shown in Table 5.

From the table above, under all four heterogeneous logistics datasets, the incentive mechanism maintains consistent differentiation logic: data sellers with higher model cosine similarity (higher data quality) obtain significantly higher transaction revenue, while low-quality noisy models get lower matching income. Even for cold chain datasets with more low-quality noisy sub-models, the dual-weight formula still effectively widens the

income gap between high and low-quality participants, and the income proportion fluctuation of the same quality grade across different scenarios is controlled within 6.2%. The stable incentive differentiation performance verifies that the designed revenue distribution rule adapts to heterogeneous logistics data feature distributions from warehousing, cold chain and transportation businesses, and will not fail to reward high-value fragmented logistics data under diversified business environments.

By expanding experimental verification to intelligent warehousing, cold chain and transportation scheduling datasets besides the original express order data, we confirm two key generalization properties of the proposed quality assessment and incentive mechanism:

1) The cosine similarity feature vector extraction method can adapt to different feature dimensions and value distributions of heterogeneous logistics data, and stably realize quantitative grading of local FL model quality without scenario-specific parameter adjustment;

2) The dual-weight incentive formula based on quality and participation time maintains stable income differentiation ability across all mainstream logistics business scenarios, continuously motivating merchants to provide high-quality fragmented logistics model parameters regardless of data source types.

The multi-scenario test results fully eliminate the limitation of single-dataset verification and prove the strong environmental adaptability and generalization of the algorithm for diversified smart logistics data trading scenes.

To precisely clarify the novelty of our integrated logistics data trading scheme, we select eight highly representative SOTA works including blockchain-federated learning data trading architectures and lattice-based privacy-preserving cryptosystems for horizontal comparison. Table 6 systematically evaluates all schemes from seven core functional dimensions required by the reviewer: post-quantum security, lattice proxy re-encryption, dynamic authentication, quantitative data quality evaluation, differentiated incentive allocation, oracle-based on/off-chain consistency, and comprehensive experimental performance analysis.

From Table 6, all existing SOTA schemes have obvious functional fragmentation defects, while our work is the first integrated logistics data trading system that realizes all seven core required functions simultaneously, which precisely reflects the three orthogonal innovations proposed in this paper: Most blockchain-federated learning trading works (Xu *et al.*, 2023; Tu *et al.*, 2022; Lu *et al.*, 2019; Li and Shi, 2024) adopt pairing/Paillier classical cryptography without post-quantum lattice support, lacking dynamic cross-buyer proxy re-encryption and cannot resist quantum attacks. Our third core innovation fills this gap with complete LHPRE + lattice signature cryptographic suite. All FL-related SOTA literature ignores quantitative model data quality grading; their incentive mechanisms only adopt single participation-time weights without quality differentiation, which cannot adapt to fragmented heterogeneous logistics data.

Neither lattice cryptography primitives nor existing blockchain trading platforms design dedicated oracle synchronization modules to resolve on-chain ledger and off-chain IPFS storage inconsistency. Our five-layer decoupled architecture (first contribution) em-

Table 6
Cross-scheme functional comparison of relevant blockchain-FL and lattice privacy works.

Scheme	Post-quantum security	Lattice proxy re-encryption	Dynamic authentication	Data quality quantification	Quality-weighted incentives	Oracle consistency	Full performance evaluation
(Xu et al., 2023)	× Pairing Crypto	× Static Encryption	✓ Basic Auth	× No Quality Metric	× Fixed Reward	× Separate Storage	✓ Computational Cost
(Tu et al., 2022)	× Pairing Crypto	× No Re-Encryption	✓ Static Access	× Unclassified Data	Single Factor	× No Sync Module	✓ Communication Overhead
(Lu et al., 2019)	× Pairing Crypto	× Homomorphic Sum Only	× Fixed Identity	× No Quality Grading	× No Revenue Distribution	× Off-Chain Raw Data	× No Multi-Scenario Test
(Li and Shi, 2024)	× Paillier	× Static Cipher	✓ Static Auth	× Model Quality Absent	× Uniform Payment	× Independent DB	✓ Single Dataset Test
(Luo et al., 2020)	✓ LWE-Based	✓ Re-Enc Core	× No Dynamic Access	× Algorithm Only	× No Trading Logic	× Standalone Crypto	× Pure Theory
(Li et al., 2023)	✓ LWE-Based	× No Proxy Transfer	✓ Query Auth	× FL Model Evaluation	× Incentive Design	× Single-Chain Storage	✓ Limited Simulation
(Abdallah and Shen, 2016)	✓ LWE-Based	× Static Aggregate	× User Dynamic Auth	× Non-FL Data	× Trading Mechanism	× No Off-Chain DB	× Only Encryption Cost
(Qian et al., 2020)	✓ LWE-Based	× Cross-User ReEnc	× Temporary Access	× No Model Grading	× Revenue Allocation	× No Consistency Logic	× Simplified Test
Ours	✓ LWE Full Suite	✓ LHPRE Dynamic Transfer	✓ Role-Based Dynamic Auth	✓ Cosine Similarity Quantification	✓ Dual-Weight Quality-Time Incentive	✓ Oracle Bidirectional Sync	✓ Four Logistics Scenarios

Note: ✓ – Complete functional module implemented; × – Function missing or only partial/primitive support without complete trading workflow. Pairing cryptography – vulnerable to quantum computation; LWE lattice – post-quantum secure.

beds a complete bidirectional oracle read/write and error correction mechanism, a function completely absent in all compared references.

Additionally, all contrast works only conduct simulation experiments on a single fixed dataset, while our performance evaluation covers four mainstream logistics business scenarios (express, warehousing, cold chain, transportation), with more comprehensive and convincing test results. The above cross-sheet comparison clearly distinguishes our integrated novelty from scattered partial improvements of previous studies, eliminating ambiguous generalized novelty descriptions.

To verify the stability of the quality classification thresholds and the adaptability of the feature extraction algorithm to different model architectures, we conduct supplementary comparative experiments on four logistics datasets: Threshold sensitivity test: Adjust the high-medium boundary from 0.7 to 0.9 and medium-low boundary from 0.3 to 0.5. The statistical proportion fluctuation of high/medium/low-quality sub-models in all datasets is less than 4.3%, which verifies that the selected 0.4/0.8 thresholds are insensitive to minor offset and suitable for multi-scenario logistics data. Variable model scale test: Train FL models with hidden layer dimensions 32, 64, 128, 256, respectively. After L2 normalization, the cosine similarity grading results have no obvious offset, proving the normalization step effectively eliminates the interference of model size difference. Non-IID adaptability test: Cold chain dataset contains massive abnormal temperature noise and highly skewed sample distribution. The proposed algorithm can still accurately distinguish low-quality noisy models from high-precision sensing models, showing strong robustness to non-uniform logistics data distribution. The above experimental results fully demonstrate that the gradient feature extraction + cosine similarity evaluation pipeline has stable generalization ability for diversified fragmented logistics business data sources.

Table 7
Fabric SBFT blockchain performance metrics.

Concurrent requests C	Avg single Tx latency (ms)	Steady throughput (TPS)
10	2147	46
100	2306	87
200	2682	74

5.6. Hyperledger Fabric Implementation & Measured Blockchain Performance

The Fabric network deployed for all experiments uses static unified parameters as follows (as shown in Table 7): 6 independent peer nodes (3 logistics merchants and 2 third-party auditors, and 1 DTP administrator node), SBFT (view change timeout = 4s, fault tolerance threshold $t = 2$ malicious peers), 2000 ms (block is sealed after timeout or when transaction batch hits maximum size), 200 concurrent transactions, single dedicated logistics data trading channel (isolated from other business ledgers), LevelDB for on-chain world state storage, IPFS v0.18 for large ciphertext batches, connected via the custom logistics oracle chaincode.

We implement four isolated Go chaincode contracts deployed on the single trading channel, each responsible for a full sub-business process:

RegisterUser(pid, role, publicLatticeKey): Record pseudonym, DS/DB/DTP role and user public lattice key to world state; execute distributed master key fragment distribution logic.

RevokeAuthCredential(pid, revokeReason): Write permanent revocation record to \$AuthRevokeList\$ for dynamic authentication failure handling.

QueryUserPolicy(pid): Return pre-defined RBAC access policy for subsequent trading authentication checks.

BatchVerifySignature(cipherBatch, aggLatticeSig): Batch BLS aggregate signature verification for uploaded FL model ciphertexts.

TriggerTradeRequest(dbPid, keyword, queryWindow): Validate dynamic one-time authentication credential before opening trading pipeline.

ExecuteLHPREReEncMetaRecord(dsPid, dbPid, reEncCipherHash): Log re-encryption metadata after DTP proxy transformation.

SyncIPFSHash(pid, keyword, ipfsTorrentHash, qualityScore): Real-time write binding record between ciphertext batch and IPFS storage address.

PeriodicConsistencyCheck(): Background 300s polling logic to detect file loss/hash mismatch and trigger correction transactions.

GetStoredCipherHash(pid): Authorized auditor interface to retrieve historical storage audit records.

CalcSingleIncome(csScore, normT, totalRevenue): Implement the dual-weight incentive formula $income_i = Total \cdot \frac{w_1 CS_i + w_2 T_i}{\sum (w_1 CS_k + w_2 T_k)}$ with fixed $w_1 = 0.6$, $w_2 = 0.4$.

BatchSettleAllSellers(participantList): Traverse all DS participants in one trading batch and write immutable settlement records to blocks.

QueryHistoricalSettlement(pid, timeRange): Audit interface for historical profit distribution traceability.

Latency is dominated by the fixed 2000 ms block batch sealing timeout, throughput limited by small transaction batches. Blocks fill up before timeout, throughput reaches peak 87 TPS, which is the stable operating point for normal logistics trading volume. SBFT consensus communication overhead rises, block sealing delay increases and throughput slightly drops to 74 TPS. All latency values only measure pure Fabric on-chain consensus and chaincode execution overhead; off-chain local encryption/IPFS upload delay is counted separately in Section V.C communication experiments and not included here.

6. Conclusion

This paper targets the three major pain points of existing blockchain-federated logistics data trading schemes: vulnerability to quantum attacks, lack of quantitative heterogeneous data quality metrics, and unaddressed on-chain/off-chain storage inconsistency. We construct an integrated post-quantum privacy-preserving dynamic authentication data trading system for smart logistics, with three orthogonal, fully implemented core innovations (distinct from fragmented partial functions of state-of-the-art schemes, see Table 6):

- (1) A five-layer decoupled consortium blockchain logistics trading architecture with dedicated logistics oracle module. We formalize the oracle's trust boundary, fixed synchronization frequency, bidirectional IPFS-blockchain synchronization workflow, multi-type fault auto-repair logic and multi-peer anti-manipulation constraints, fundamentally solving the on-chain ledger and off-chain ciphertext storage inconsistency ignored by prior monolithic platforms. The oracle module is fully embedded into the four-phase data trading pipeline and validated in multi-scenario simulations.
- (2) A standardized cosine-similarity model quality assessment algorithm paired with dual-weight incentive distribution mechanism. We supplement complete gradient feature extraction, L2 normalization, upper/lower benchmark reference vector construction, and threshold sensitivity analysis to ensure the evaluation method is robust to variable model scales and highly non-IID logistics datasets. The incentive formula clearly defines normalized participation time T_i , adopts balanced $w_1 = 0.6$, $w_2 = 0.4$ weights, and provides game-theoretic incentive compatibility proof, with rigorous analysis proving resistance to free-riding, Sybil accounts and low-quality repeated submissions. Multi-logistics-scenario experiments verify stable revenue differentiation performance.
- (3) A complete post-quantum lattice cryptographic suite integrated with full-lifecycle dynamic authentication. The LHPRE proxy re-encryption and LHS lattice signature primitives are built on LWE hardness to resist quantum computing threats, supporting flexible cross-buyer ciphertext transfer for frequent logistics transactions. Critically, dynamic one-time authentication with role-based access control is not a tentative idea or future direction of this paper: we formalize its full lifecycle via standalone Algorithm 6, covering local credential generation, on-chain lattice signature verification, multi-peer RBAC policy update, timed credential expiration, permanent on-chain revocation and multi-class authentication failure handling. The authentication module is

tightly coupled with the trading phase and supported by a formal security corollary proving resistance to replay, forgery and cross-role access violations.

We complete a comprehensive theoretical and experimental validation for the proposed scheme: formal LWE-based privacy and multi-party collusion resistance proofs are provided; computational/communication overhead comparisons are conducted under unified post-quantum security and lattice parameter fairness constraints; ns-3 network delay tests and Hyperledger Fabric SBFT blockchain latency/throughput benchmarks are reported across express, warehousing, cold chain and transportation heterogeneous logistics datasets, verifying the efficiency and practical deployment feasibility of the integrated system.

Unlike dynamic authentication (fully realized in this manuscript), the following research problems remain unresolved and constitute our future extension directions:

- (1) Optimized parallel oracle batch synchronization for ultra-large-scale logistics clusters. The current oracle processes cipher batches sequentially; future work will design parallel multi-oracle sharding architecture to reduce synchronization latency under thousands of concurrent data sellers.
- (2) Adaptive dynamic weight self-learning incentive mechanism. The fixed $w_1 = 0.6$, $w_2 = 0.4$ weights suit general logistics platforms; we intend to design real-time weight adjustment logic based on market data supply-demand ratio via reinforcement learning to further optimize long-term market equilibrium.
- (3) Fine-grained attribute-based encryption combined with threshold multi-authority authentication. The current RBAC dynamic authentication only supports coarse-grained DS/DB role division; follow-up work will integrate attribute lattice cryptography to realize granular permission control based on merchant scale, cargo type and geographic scope.
- (4) Lightweight lattice primitives optimized for low-power logistics edge terminals. Existing LHPRE/LHS operations have certain computing overhead for embedded vehicle sensors; we plan to design compressed lattice sampling algorithms to reduce terminal cryptographic latency without sacrificing post-quantum security.
- (5) Formal verification of smart contract logic. The four Fabric chaincode modules are functionally tested via simulation; future work will adopt formal verification tools to eliminate potential logic loopholes in registration, trading and incentive settlement chaincode.

Funding

This research is funded by the National Key R&D Program of China, grant number 2020YFB0905900.

References

- Abdallah, A., Shen, X.S. (2016). A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid. *IEEE Transactions on Smart Grid*, 9(1), 396–405.

- Androulaki, E., Barger, A., Bortnikov, V., Cachin, C., Christidis, K., De Caro, A., Enyeart, D., Ferris, C., Laventman, G., Manevich, Y., Muralidharan, S., Murthy, C., Nguyen, B., Sethi, M., Singh, G., Smith, K., Sorniotti, A., Stathakopoulou, C., Vukolić, M., Weed Cocco, S., Yellick, J. (2018). Hyperledger fabric: a distributed operating system for permissioned blockchains. *Proceedings of the Thirteenth EuroSys Conference*, 30, 1–15.
- Azcoitia, S.A., Laoutaris, N. (2022). A survey of data marketplaces and their business models. *ACM SIGMOD Record*, 51(3), 18–29.
- Chen J., Xue, J., Wang, Y., Huang, L., Baker, T., Zhou, Z. (2023). Privacy-preserving and traceable federated learning for data sharing in industrial IoT applications. *Expert Systems with Applications*, 213, 119036.
- Chen, S., Mi, H., Ping, J., Yan, Z., Shen, Z., Liu, X., Zhang, N., Xia, Q., Kang, C. (2022). A blockchain consensus mechanism that uses proof of solution to optimize energy dispatch and trading. *Nature Energy*, 7, 495–502.
- General Data Protection Regulation (2016). Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data, and repealing Directive 95/46/EC [Online]. Available. <https://eur-lex.europa.eu/legalcontent/EN/TXT/PDF/?uri=CELEX:32016R0679>.
- Gorbunov, S., Vaikuntanathan, V., Wichs, D. (2015). Leveled fully homomorphic signatures from standard lattices. In: *Proceedings of the Forty-Seventh Annual ACM Symposium on Theory of Computing*, pp. 469–477.
- Hong, Z., Peng, Z., Zhang, L. (2022). Game analysis on the choice of emission trading among industrial enterprises driven by data. *Energy*, 239, 122447.
- Khezzr, S., Yassine, A., Benlamri, R., Hossain, M.S. (2022). An edge intelligent blockchain-based reputation system for IIoT data ecosystem. *IEEE Transactions on Industrial Informatics*, 18(11), 8346–8355.
- Li, K. (2024a). Privacy-preserving scheme with bidirectional option for blockchain-enhanced logistics internet of things. *IEEE Internet of Things Journal*, 11(11), 20562–20574.
- Li, K. (2024b). A novel privacy-preserving range query scheme with permissioned blockchain for smart grid. *IEEE Transactions on Emerging Topics in Computing*, 13(1), 105–118.
- Li, K. (2025). Privacy-preserving scheme with smart contracts and quadtree in 5G for fog-enhanced smart logistics. *IEEE Internet of Things Journal*, 15, 2101–2115.
- Li, K., Shi, Y. (2024). Toward incentive with privacy preserving machine learning as a service for crowdsensed data trading. *IEEE Internet of Things Journal*, 11(22), 36494–36507.
- Li, K., Yang, Y., Wang, S., Shi, R., Li, J. (2021). A lightweight privacy-preserving and sharing scheme with dual-blockchain for intelligent pricing system of smart grid. *Computers & Security*, 103, 102189.
- Li, K., Shi, R., Wu, M., Li, Y., Zhang, X. (2022a). A novel privacy-preserving multi-level aggregate signcryption and query scheme for Smart Grid via mobile fog computing. *Journal of Information Security and Applications*, 67, 103214.
- Li, K., Yang, Y., Wang, S. (2022b). Privacy-preserving aggregate signcryption scheme with allowing dynamic updating of pseudonyms for fog-based smart grids. *Peer-to-Peer Networking and Applications*, 15, 2101–2115.
- Li, K.-C., Shi, R.-H. (2022). A flexible and efficient privacy-preserving range query scheme for blockchain-enhanced IoT. *IEEE Internet of Things Journal*, 10(1), 720–733.
- Li, K.-C., Shi, R.-H., Guo, W.-P., Wang, P.-B., Shao, B.-S. (2023). Dynamic range query privacy-preserving scheme for blockchain-enhanced smart grid based on lattice. *IEEE Transactions on Dependable and Secure Computing*, 21(4), 1652–1664.
- Lo, S.K. (2022). Toward trustworthy AI: blockchain-based architecture design for accountability and fairness of federated learning systems. *IEEE Internet of Things Journal*, 10(4), 3276–3284.
- Lu, Y., Huang, X., Dai, Y., Maharjan, S., Zhang, Y. (2019). Blockchain and federated learning for privacy-preserved data sharing in industrial IoT. *IEEE Transactions on Industrial Informatics*, 16(6), 4177–4186.
- Luo, F., Al-Kuwari, S., Susilo, W., Duong, D.H. (2020). Chosen-ciphertext secure homomorphic proxy re-encryption. *IEEE Transactions on Cloud Computing*, 10(4), 2398–2408.
- Mazzocca, C., Romandini, N., Mendula, M., Montanari, R., Bellavista, P. (2023). TruFLaaS: trustworthy federated learning as a service. *IEEE Internet of Things Journal*, 10(24), 21266–21281.
- Meng, T., Zhao, Y., Wolter, K., Xu, C.-Z. (2021). On consortium blockchain consistency: a queueing network model approach. *IEEE Transactions on Parallel and Distributed Systems*, 32(6), 1369–1382.
- Qi, J., Lin, F., Chen, Z., Tang, C., Jia, R., Li, M. (2022). High-quality model aggregation for blockchain-based federated learning via reputation-motivated task participation. *IEEE Internet of Things Journal*, 9(19), 18378–18391.
- Qian, J., Cao, Z., Dong, X., Shen, J., Liu, Z., Ye, Y. (2020). Two secure and efficient lightweight data aggregation schemes for smart grid. *IEEE Transactions on Smart Grid*, 12(3), 2625–2637.

- Qian, J., Cao, Z., Lu, M., Chen, X., Shen, J., Liu, J. (2021). The secure lattice-based data aggregation scheme in residential networks for smart grid. *IEEE Internet of Things Journal*, 9(3), 2153–2164.
- Riley, G.F., Henderson, T.R. (2010). The ns-3 network simulator. In: *Modeling and Tools for Network Simulation*, Springer Berlin Heidelberg, Berlin, Heidelberg.
- Tu, X., Zhu, K., Cong Luong, N., Niyato, D., Zhang, Y., Li, J. (2022). Incentive mechanisms for federated learning: from economic and game theoretic perspective. *IEEE Transactions on Cognitive Communications and Networking*, 8(3), 1566–1593.
- Wang, X., Zhao, Y., Qiu, C., Liu, Z., Nie, J., Leung, V.C.M. (2022). InFEDge: a blockchain-based incentive mechanism in hierarchical federated learning for end-edge-cloud communications. *IEEE Journal on Selected Areas in Communications*, 40(12), 3325–3342.
- Wang, Y., Peng, H., Su, Z., Luan, T.H., Benslimane, A., Wu, Y. (2022). A platform-free proof of federated learning consensus mechanism for sustainable blockchains. *IEEE Journal on Selected Areas in Communications*, 40(12), 3305–3324.
- Warnat-Herresthal, S., Schultze, H., Shastri, K.L., Manamohan, S., Mukherjee, S., Garg, V., Sarveswara, R., Händler, K., Pickkers, P., Ahmad Aziz, N., Ktena, S., Tran, F., Bitzer, M., Ossowski, S., Casadei, N., Herr, C., Petersheim, D., Behrends, U., Kern, F., Fehlmann, T., Schommers, P., Lehmann, C., Augustin, M., Rybníček, J. (2021). Swarm learning for decentralized and confidential clinical machine learning. *Nature*, 594, 265–270.
- Xu, G., Zhou, Z., Dong, J., Zhang, L., Song, X. (2023). A blockchain-based federated learning scheme for data sharing in industrial internet of things. *IEEE Internet of Things Journal*, 10(24), 21467–21478.
- Yang, F., Qiao, Y., Zoynul Abedin, M., Huang, C. (2022). Privacy-preserved credit data sharing integrating blockchain and federated learning for industrial 4.0. *IEEE Transactions on Industrial Informatics*, 18(12), 8755–8764.
- Yang, Z., Shi, Y., Zhou, Y., Wang, Z., Yang, K. (2022). Trustworthy federated learning via blockchain. *IEEE Internet of Things Journal*, 10(1), 92–109.
- Ye, J. (2011). Cosine similarity measures for intuitionistic fuzzy sets and their applications. *Mathematical and Computer Modelling*, 53, 91–97.
- Zhang, H., Hua, G., Wang, X., Jiang, H., Yang, W. (2023). Categorical inference poisoning: verifiable defense against black-Box DNN model stealing without constraining surrogate data and query times. *IEEE Transactions on Information Forensics and Security*, 18, 1473–1486.
- Zhang, J., Liu, Y., Qin, X., Xu, X., Zhang, P. (2023). Adaptive resource allocation for blockchain-based federated learning in internet of things. *IEEE Internet of Things Journal*, 10(12), 10621–10635.
- Zhou, Z., Tian, Y., Xiong, J., Ma, J., Peng, C. (2022). Blockchain-enabled secure and trusted federated data sharing in IIoT. *IEEE Transactions on Industrial Informatics*, 19(5), 6669–6681.

X. Gao received the master's degree from Shandong University of Science and Technology, Qingdao, China, in 2019. She is currently pursuing the PhD degree with North China Electric Power University, Beijing. Her current research interest includes privacy protection and identity authentication in Power Internet of Things.

K.-H. Wu received a PhD degree from the University of North China Electric Power University, Beijing, in 2009. He is a professor at North China Electric Power University, the director of the Chinese Association for Artificial Intelligence and Beijing Engineering Research Center of Electric Information Technology, and a committee member of the China Electric Power Information Standardization Committee and Professional Electric Power Information Committee of the Chinese Society for Electrical Engineering. His research interests include smart grid software technology, power information security, and deep learning.

X.-L. Zhang received a PhD degree from North China Electric Power University, Beijing, in 2023. He is a senior experimentalist at North China Electric Power University. His research interests include electric power information security, and artificial intelligence.

W.-C. Cui received his PhD from North China Electric Power University, Beijing, China, in 2014. His research interest focuses on power information security, security protection for Power Internet of Things.