Energy Data Transaction Privacy Protection Scheme Based on Dynamic Pseudonym and Lightweight zk-SNARKs

Rui Xin[1], ShaoYing Wang[1], Xin Lu[1], YanYan Lu[1], LiPing Yang[1] and XinYing Wang[2,*]

[1]*State Grid Hebei Electric Power Company Information and Communication Branch, Shijiazhuang City, Hebei Province 050000, China*
[2]*Department of Computer Science, North China Electric Power University, Baoding City, Hebei Province 071003, China*
E-mail: wangxinying@ncepu.edu.cn
[*]*Corresponding Author*

Received 05 December 2025; Accepted 03 March 2026

Abstract

In response to the difficulty of balancing privacy protection and system efficiency in energy data trading, this article analyzes the limitations of existing methods: static pseudonym mechanisms can easily lead to long-term identity link risks, traditional zk-SNARKs schemes have high computational overhead, and Raft consensus mechanisms lack robustness in adversarial environments. To address the above challenges, an integrated privacy protection scheme based on dynamic pseudonyms and lightweight zk-SNARKs is proposed. This scheme breaks the temporal correlation of transactions

Project Funding: Science and Technology Project of State Grid Hebei Electric Power Co., Ltd. (Research on Key Supporting Technologies for Value-Added Services of Energy Data Elements: kj2023-038)

Journal of Cyber Security and Mobility, Vol. 15_3, 653–682.
doi: 10.13052/jcsm2245-1439.1536

through a dynamic pseudonym generation mechanism, uses blockchain level batch processing proofs to reduce the computational and storage overhead of zero knowledge proofs, and introduces an LSTM based node health assessment model and incremental log synchronization mechanism to enhance the error tolerance and synchronization efficiency of the Raft consensus algorithm. The experimental results show that the proposed scheme outperforms traditional methods in terms of privacy, transaction processing performance, and system availability, effectively achieving a balance between privacy protection and operational efficiency, and providing a feasible technical path for energy data trading.

Keywords: Zero-knowledge proof, privacy protection, raft consensus algorithm, dynamic pseudonym, energy data.

1 Introduction

Driven by the deep integration of the Internet of Things (IoT), the Industrial Internet, and blockchain technology, the scale of cross-domain energy trading data transactions is continuously expanding. In this context, balancing privacy protection with transaction efficiency has emerged as a critical bottleneck hindering industrial adoption. While the inherent properties of blockchain – such as its immutability, and decentralized and distributed nature – provide a trusted platform for executing these transactions, traditional blockchain architectures exhibit significant shortcomings in identity anonymity, the efficiency of privacy-preserving proofs, and consensus robustness. For instance, static identifiers can easily enable attackers to trace transaction trajectories. The substantial computational and storage overhead associated with zero-knowledge proofs (zk-SNARKs) renders them unsuitable for high-frequency trading. Furthermore, the classic Raft consensus algorithm is vulnerable to service disruptions under Byzantine attacks and in wide-area network (WAN) environments. In the energy data trading scenario, trading has significant high-frequency characteristics, such as real-time load data exchange in the electricity spot market often reaching second or minute level frequencies, the data granularity covers from fine-grained user side metering data (such as electricity consumption at 15 minute intervals) to aggregated regional statistical information, the participants include power generation companies, grid companies, power sales companies, and end users, and their trust model is usually a partially trusted consortium chain environment – there is commercial competition between nodes but they need

to follow unified transaction rules. Therefore, there are both potential malicious nodes in the system (such as trying to obtain competitor information by analyzing transaction flows), as well as non-Byzantine node failures caused by network fluctuations or hardware failures.

Existing dynamic pseudonym solutions are often tailored to specific application scenarios, lacking the flexibility required for multi-source heterogeneous energy data transactions. Zhang et al. [1] proposed a fog computing-based scheme (DPSP) to reduce management latency and enhance resistance against correlation attacks in the Internet of Vehicles. However, their work focuses on location privacy and does not address the protection of identity and transaction content. Furthermore, Li et al. [2] introduced a pseudonym replacement scheme (TLAS) using traffic context prediction, yet it relies on static mixed zones in traffic scenarios and cannot adapt to the needs of data transactions. In a different approach, Wang et al. [3] developed a partial pseudonym ID-based key wireless generation algorithm for RFID systems, achieving coordinated control of pseudonyms and keys. Nevertheless, this algorithm targets perception-layer devices and is unsuitable for dynamic identity updates in data transactions. Overall, these existing schemes have not established an identity dynamic update mechanism suitable for multi-source data transactions, resulting in rigid replacement strategies that fail to accommodate highly active users.

Research on lightweight zk-SNARKs has primarily focused on memory optimization, scenario adaptation, and scalability. However, a comprehensive solution that harmonizes computational efficiency with broad applicability remains an open challenge. Qi et al. [4] proposed a hash-based memory optimization method (Split) to mitigate zk-SNARK circuit storage overflow, yet their work narrowly targets single-circuit optimization without addressing batch-proof efficiency for multi-transaction environments. Luong et al. [5] leveraged zk-SNARKs for anonymous health data access and encryption, but their data structure's excessive proof-generation latency renders it impractical for high-frequency transactions. Guan et al. [6] introduced BlockMaze, an account-model privacy scheme to obscure balances and transaction amounts, although it overlooks critical optimizations for on-chain proof storage overhead. Luong et al. [7] later applied zk-SNARKs to blockchain identity management for selective attribute disclosure, yet their design lacks mechanisms to aggregate proofs in multi-transaction workflows.

Further limitations emerge in scalability-focused efforts. Wang et al. [8] devised a cross-chain transaction scheme combining multi-signatures and zero-knowledge proofs to bolster privacy, but their work neglects intra-chain

proof efficiency. Li [9] systematically analyzed zero-knowledge proofs in blockchain privacy, identifying computational complexity as the central impediment to adoption. Wu et al. [10] proposed a homomorphic encryption-backed zk-SNARK variant for scalability, but this introduces prohibitive key-management complexity. Collectively, these efforts reveal a persistent trade-off: existing solutions either optimize isolated performance dimensions (e.g., memory or single-scenario throughput) or bind tightly to niche applications. This dichotomy obstructs the realization of zk-SNARKs that meet the trifecta of low-latency computation, minimal storage footprint, and cross-scenario versatility – essential for scalable data transactions [11].

In terms of improvements to the Raft consensus, optimizations have been made around Byzantine fault tolerance and throughput improvement, but the adaptability is insufficient in terms of data transactions. Raft is the mainstream consensus mechanism for consortium chains, but traditional Raft can only tolerate less than 50% faulty nodes, and the leader election delay is as high as 200 ms [12]. Li et al. [13] proposed CB-Raft, a Byzantine fault-tolerant variant that isolates malicious nodes through comprehensive status evaluation, significantly boosting attack resistance. While effective in node filtering, their approach neglects bandwidth optimization for log synchronization in wide-area networks, a critical gap for distributed data transactions. Li et al. [14] advanced RB-Raft, dynamically adjusting voting weights to enhance fault tolerance against Byzantine nodes. However, their model assumes static node conditions, overlooking the real-time status fluctuations inherent in high-frequency data exchanges. Li et al. [15] designed a multi-master Byzantine fault-tolerant Raft mechanism to improve consensus throughput, but the coordination logic between master nodes was complex, resulting in increased latency. Traditional Raft and its improved solutions either focus on fault tolerance or adapt to specific scenarios but have not formed a consensus mechanism adapted to data transactions. This leads to large fluctuations in data transaction confirmation delays and a high risk of service interruption under Byzantine attacks.

In summary, existing research exhibits critical limitations in three key dimensions: the scenario-specific adaptability of dynamic pseudonyms, the computational overhead trade-offs in lightweight zk-SNARKs, and the seamless integration of enhanced Raft consensus mechanisms. In order to bridge these gaps, this article introduces an integrated privacy protection framework for data transactions, which includes the following three innovative points: firstly, to address the problem of long-term behavior tracing caused by static

pseudonyms in high-frequency transaction scenarios, a dynamic pseudonym generation mechanism based on transaction content and timestamp combined perturbation is proposed, which achieves strong decoupling between pseudonyms and user identities; secondly, to address the issue of high computational overhead in traditional zk-SNARK single proof generation models in high concurrency environments, a block level batch proof generation and aggregation verification method has been designed, significantly reducing the computational and storage costs per transaction; thirdly, in response to the insufficient response of Raft consensus to node state fluctuations in partially trusted environments, an LSTM based node health dynamic evaluation model and incremental log synchronization mechanism were introduced to enhance the fault tolerance and synchronization efficiency of the consensus algorithm. By integrating these methods, the proposed solution not only addresses previous shortcomings, but also lays a solid foundation for the application of blockchain in energy data transactions, promoting the collaborative development between privacy preserving computing and distributed consensus technology.

2 Related Technical Foundation

2.1 Pseudonymization Technology

Pseudonymization serves as a core privacy-enhancing technology, the implementation principle of which involves substituting users' real identifiers with pseudonymous ones, thereby concealing their true identity to enable secure data exchange and reliable user services. It finds primary application in domains such as location-based services (LBS), mobile crowdsourcing, and electronic payments, where it establishes a robust barrier between user identity and behavior. Pseudonymization strategies are broadly categorized into static and dynamic approaches, which differ substantially in their replacement frequency, the level of privacy assurance they provide, and their respective suitability for various application scenarios.

2.1.1 Static pseudonymization technology

Pseudonymization is broadly categorized into static and dynamic approaches. Static pseudonym technology refers to a privacy-preserving method in which users employ a fixed pseudonym throughout the service duration. Its core characteristic lies in the stable mapping between the pseudonym and the real identity, with the replacement typically occurring only once during the

initial registration phase. This approach is typically managed in a centralized manner, where a central authority assigns a pseudonym to each user, resulting in low system overhead. However, this method has limited applicability and is primarily suitable for low-frequency, short-term service scenarios, such as one-time query requests. A more significant drawback is its inherent weakness in privacy protection. Since users employ a fixed pseudonym over an extended period, it creates a long-term, traceable behavioral trajectory. Malicious actors can leverage this persistence, correlating it with auxiliary background knowledge of the user's temporal and spatial activities, to ultimately link the pseudonym back to the real-world identity.

2.1.2 Dynamic pseudonym technology

Dynamic pseudonym technology represents an advanced solution designed to mitigate the inherent limitations of static pseudonyms. Its core objective is to disrupt the long-term linkage between user identity and behavior through the periodic rotation of pseudonyms and their spatial isolation. This is achieved through the use of "mixing zones" – designated areas or logical domains where the system mandates a pseudonym update upon a user's entry. Within these zones, the pseudonyms of multiple users are collectively reassigned, creating a complex, many-to-many mapping relationship that effectively obscures traceability. A schematic of this pseudonym rotation process is depicted in Figure 1.

The update frequency of pseudonyms is intrinsically linked to the distribution of mixing zones, which prevents users from retaining the same pseudonym over extended periods and thereby provides robust anti-tracking capability. To illustrate, as shown in the figure, four users can generate up to 24 possible pseudonym replacement scenarios. As the number of users n grows, the number of possible mapping combinations increases factorially $(n!)$, making it computationally infeasible for an attacker to recover real identities by tracking pseudonyms. Although dynamic pseudonym algorithms significantly enhance anti-tracking through mixing zones and periodic replacement mechanisms, they still face challenges in real-world complex scenarios, such as achieving low-latency responses, establishing reliable trust mechanisms, and enabling secure data collaboration. Therefore, to realize the dual optimization of privacy and performance, these algorithms often need to be deeply integrated with complementary technologies – such as edge computing, blockchain, and federated learning – to form a more adaptive and resilient privacy-enhancing architecture.

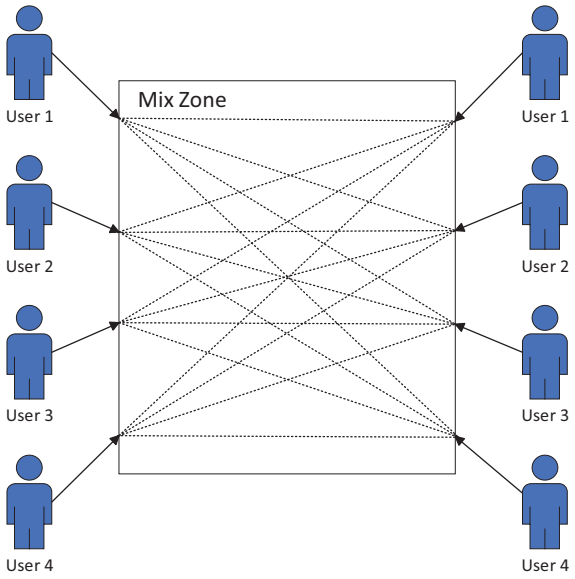

Figure 1 Schematic diagram of the pseudonym change process.

2.2 Zero-Knowledge Proof

Zero-knowledge proof is a cryptographic protocol that allows a prover to prove the truth of a statement to a verifier without revealing any additional information. A secure zero-knowledge proof possesses three core properties: integrity, soundness, and zero-knowledge. This article introduces zero-knowledge proofs in four stages.

(1) Circuit and polynomial constraints: Any computational problem can be expressed as a Boolean circuit or an arithmetic circuit. For an arithmetic circuit, the prover must prove that they know the input that satisfies the circuit ω, i.e. $C(\omega, x) = 0$, where C is the circuit function, x is the public input, ω, and s is the secret evidence. Circuit constraints can be converted into polynomial constraints through Lagrange interpolation. Let the polynomial corresponding to the circuit output be $p(X)$, then the proof objective is equivalent to $P(s) = 0$, where s is a random secret point.

(2) Trusted initialization: Generate a public reference string CRS, which includes the secret parameter s and a generator based on an elliptic curve $g, h \in G_1, g_t \in G_2$ (G_1, G_2 an elliptic curve group that supports pairing

operations). CRS includes:

$$CRS = \{g^{s^k}, h^{s^k}, g_t^{s^k} \mid 0 \leq k \leq d\} \tag{1}$$

where d is $p(X)$ the order of the polynomial.

(3) Proof generation: The prover uses the private evidence ω and CRS to generate the proof:

$$\pi = (A, B, C) \in G_1 \times G_2 \times G_1 \tag{2}$$

Satisfaction $A = g^{a(s)}, B = g_t^{b(s)}, C = g^{c(s)}$, among which $a(s), b(s)$, $c(s)$. It is based on $P(s) = 0$ a derived polynomial, and its specific form is determined by the circuit structure.

(4) Verification: The verifier verifies the correctness of the proof through pairing operations. The core equation is:

$$e(A, B) = e(g, g_t)^{h(s)} \cdot e(C, g_t) \tag{3}$$

The polynomial corresponding to the public input *x* is accepted if the equality holds.

2.3 Raft Consensus Algorithm

The Raft consensus algorithm is a classical for achieving consistency in distributed systems, primarily comprising three components: leader election, log replication, and log safety. In the leader election phase, if a node fails to receive heartbeat messages from the current leader within a specified timeout period, it transitions to a candidate state and initiates an election process. A node is elected as the new leader only if it secures votes from a majority of nodes in the cluster. During log replication, the leader appends client requests to its own log and then replicates them to all follower nodes. A log entry is considered committed once it has been successfully replicated to a majority of the followers, thereby ensuring consistency across the distributed system. Log safety is maintained by enforcing strict ordering of log entries and by having followers reject any AppendEntries requests that come from stale leaders. Although the Raft algorithm can tolerate failures of up to one-third of the nodes and generally addresses security and reliability concerns arising from hardware or network issues, it still exhibits certain limitations. For instance, the average latency for leader election typically exceeds 200 ms, making it difficult to meet millisecond-level responsiveness requirements in latency-sensitive scenarios. Moreover, the algorithm does not adequately account for

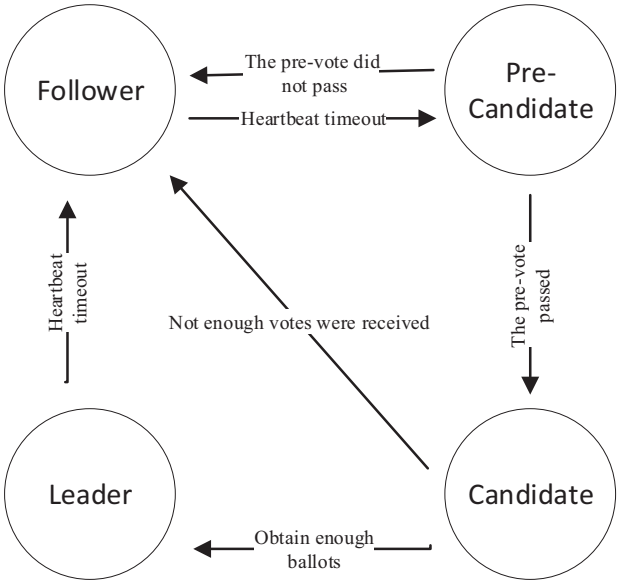

Figure 2 Flowchart of the Raft algorithm with the pre-voting mechanism.

anomalous interference in adversarial environments, which can easily lead to consensus deadlock or system unavailability.

2.3.1 Improved raft algorithm with a pre-voting mechanism

To further improve the efficiency of original Raft elections, the traditional pre-voting scheme uses a two-stage process: pre-election and formal voting, to filter out unnecessary nodes. After a follower times out, it first issues a pre-voting request as a pre-candidate. The formal election begins only after receiving approval from more than half of the nodes. Figure 2 illustrates the Raft algorithm with the pre-voting mechanism.

3 Design of a Privacy Protection Scheme Based on Dynamic Pseudonyms and Lightweight zk-SNARKs

The system architecture of this solution is built on blockchain technology, integrating a dynamic pseudonym mechanism with lightweight zero-knowledge proof (zk-SNARK) technology to achieve privacy protection and efficient processing of sensitive data transactions. The functions and interactions of each layer are shown in Figure 3.

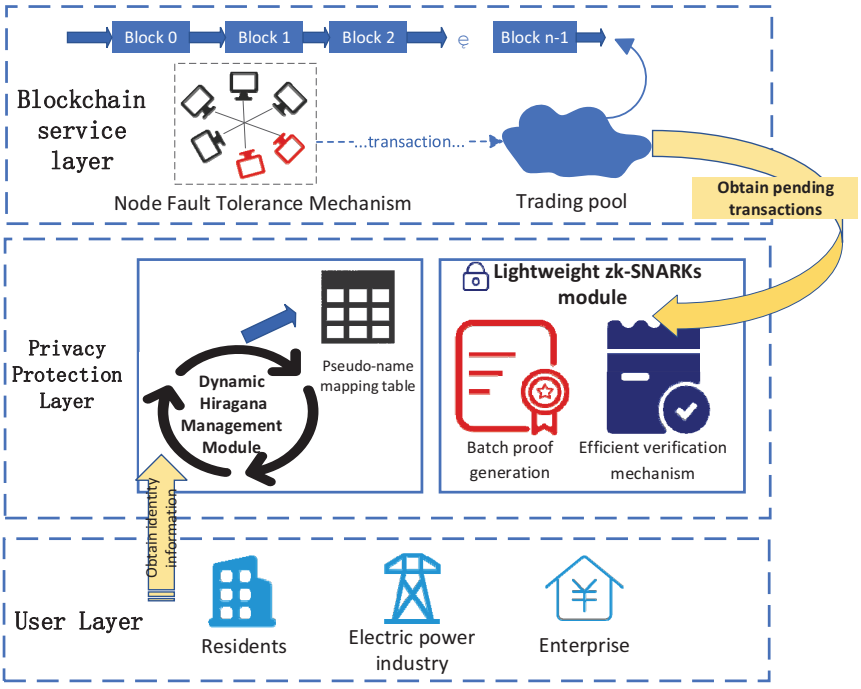

Figure 3 Architecture of a privacy protection solution based on dynamic pseudonyms and lightweight zk-SNARKs.

User layer

User entity module: This includes three types of users: residents, electric power industry, and enterprises. Each user has a unique real identity and sensitive data (such as power load, timestamp, etc.).

Transaction initiation module: The user initiates a transaction request, which includes sensitive data that needs to be uploaded, triggering the privacy protection process.

Privacy protection layer

This layer is the core of the solution and is responsible for implementing dynamic pseudonym updates and lightweight processing of zero-knowledge proofs.

The dynamic pseudonym management module maintains a dedicated pseudonym mapping table for each user. The initial pseudonym is generated

by applying a cryptographic hash function to the user's real identity. Subsequently, for each transaction, a new pseudonym is produced by combining the user's identity, a current timestamp, and associated sensitive data. This frequent updating mechanism effectively prevents long-term traceability of user activities.

Lightweight zk-SNARK module

Batch proof generation: Use block-level batch processing to reduce the cost of single transaction proof.

Efficient verification mechanism: By associating the Merkle root with the block hash, the on-chain verification process is simplified and computing resource consumption is reduced.

Blockchain service layer

Transaction pool module: temporarily stores unpackaged transactions, waiting for batch processing.

Block processing module: When the transaction pool reaches the batch threshold, it is packaged into a new block, which contains the transaction list, Merkle root, hash value of the previous block and batch zero-knowledge proof.

Node fault tolerance mechanism: This supports multi-node deployment, tolerates partial node failures through the consensus mechanism, and ensures system availability.

This hierarchical architecture effectively balances privacy protection with system performance. It leverages dynamic pseudonyms to mitigate identity tracking, employs lightweight zk-SNARKs to minimize proof overhead, and utilizes blockchain to guarantee transaction immutability and traceability. Consequently, the proposed architecture is well-suited for high-concurrency energy data trading scenarios.

3.1 Lightweight zk-SNARK Algorithm

To address the efficiency bottlenecks and privacy risks of traditional zero-knowledge proof in on-chain sensitive data transaction scenarios, this paper proposes an improved solution that combines a dynamic pseudonym mechanism with batch proof optimization.

3.1.1 Batch proof generation mechanism

Traditional zero-knowledge proof schemes require generating and verifying a separate proof for each transaction, leading to computational overhead that increases linearly with transaction volume and rendering them inefficient for high-concurrency on-chain environments. To address this, the improved scheme introduced in this paper batch-processes multiple transactions within a block into a single proof unit. These aggregated units are placed into the blockchain transaction pool, and proof generation is triggered immediately once the number of transactions in a block reaches a predefined threshold (BATCH_SIZE). Moreover, a single proof is computed for the hash set of all transactions within the block. As a result, the time complexity grows sub-linearly with the number of transactions, substantially lowering the average proof cost per transaction.

3.1.2 Collaborative privacy protection with a dynamic pseudonym and zero-knowledge proofs

To address the privacy vulnerability of traditional solutions, where user identities are long-term linked to transaction history, the improved solution deeply integrates zero-knowledge proofs with a dynamic pseudonym mechanism to form a dual privacy barrier. The zero-knowledge proof process integrating dynamic pseudonyms is shown in Figure 4.

3.2 Improved Raft Consensus Algorithm

3.2.1 Node health assessment model based on LSTM

Traditional mechanisms do not assess node health and use static thresholds. This can lead to abnormal nodes interfering with pre-voting results when attacked. This paper uses the LSTM health model to dynamically screen

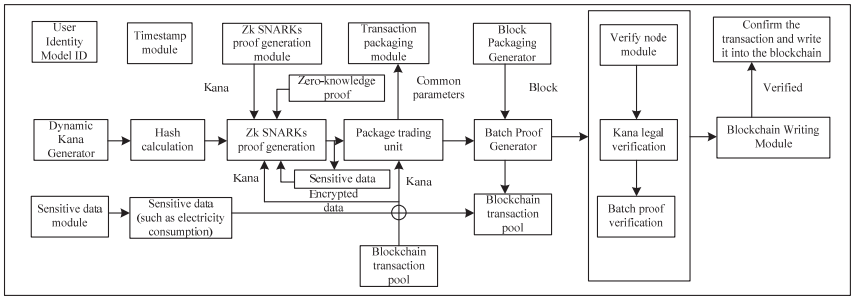

Figure 4 Zero-knowledge proof flow chart integrating dynamic pseudonyms.

pre-voting nodes and dynamically assess node health. The network structure is as follows:

Input layer: Receives node status time series data with a dimension of 50×4 (50 time steps, covering a 10-minute monitoring period; 4 types of features, including consensus performance, resource load, traffic anomalies, and historical behavior). All features are input after Min-Max normalization to eliminate dimensional interference.

Dual LSTM hidden layers: Hidden layer 1 consists of 64 LSTMs, which initially extract short-term temporal dependencies and continuously update its cell state and hidden state through updates of the input gate, forget gate, and output gate. Dropout is added to prevent overfitting. Hidden layer 2 consists of 32 LSTMs, using the output of hidden layer 1 as input to complete the extraction of long-term temporal dependencies. The calculation principle is the same as that of hidden layer 1, and Dropout is added to prevent overfitting.

Output layer: The last output of hidden layer 2 is converted to a single value through direct product, then compressed to [0, 1] through Sigmoid activation, and then scaled to produce a healthy score of [0, 100]. $S \geq 60$ points indicates a healthy node.

To train the LSTM model, node state time series data was extracted from the system's historical operation logs, and a total of 30 days of operation records were collected, including mixed scenarios of normal and attack. The tags are classified and annotated by experts based on whether the node subsequently triggers consensus anomalies or exhibits clear attack behavior: if the node causes view changes, inconsistent logs, or a sudden drop in throughput within the next 10 minutes, it is marked as unhealthy (0), otherwise it is marked as healthy (1). The training set and test set are divided in a 7:3 ratio, and the accuracy of the model on the test set reaches 94.2%, the recall rate is 92.8%, and the F1 score is 0.93.

To verify the advantages of LSTM over simple thresholding methods, it was compared with a baseline based on sliding window statistics: the baseline method calculates the mean and standard deviation within each feature window, and when any feature deviates from the mean by more than 3 times the standard deviation, it is considered unhealthy. On the same test set, the baseline method has an accuracy of only 81.5%, an F1 score of 0.79, and a false positive rate of up to 23%, while the LSTM model reduces the false positive rate to 6.5%. This indicates that LSTM can effectively capture

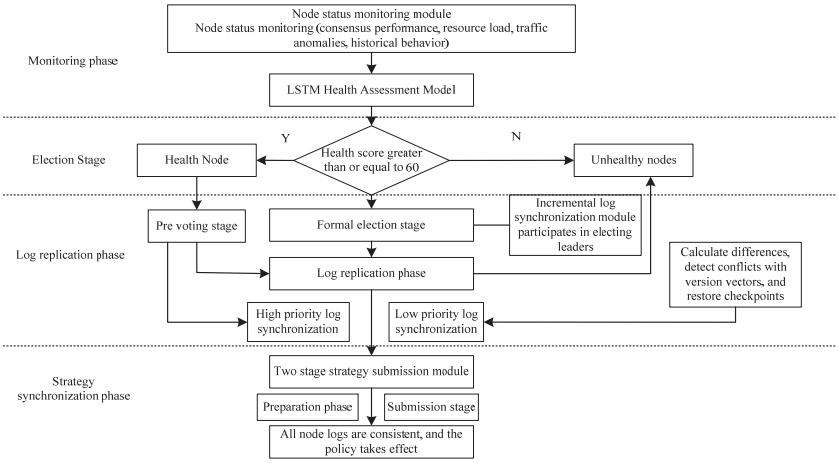

Figure 5 Improved Raft flowchart.

the temporal evolution patterns of node states significantly better than static thresholding methods. Regarding the cost of model training and updating, the model is trained offline and retrained once a week using newly added log data to adapt to the slow drift of node behavior patterns. A single training session takes about 15 minutes (using a single GPU), and the inference process is extremely lightweight, with each health assessment taking only 2.3 ms, fully meeting the latency requirements for real-time election decision-making.

Based on the LSTM health assessment, the pre-voting process is optimized as follows. Firstly, when initiating a pre-vote request, candidate nodes attach their own LSTM health time-series features. Secondly, upon receiving such a request, a node uses its local LSTM model to evaluate the candidate's health status, automatically filtering out those with a health score below 60. Finally, a candidate proceeds to the formal election phase only after receiving endorsements from over half of the healthy nodes. Concurrently, the pre-voting timeout window is extended to 0.5 s under attack scenarios to reduce failure rates. This integration of time-series modeling enhances the accuracy of health assessment, thereby ensuring both consensus efficiency and system fault tolerance.

The flowchart of the improved Raft algorithm is presented in Figure 5.

3.2.2 Incremental log synchronization and policy consensus

To meet the bandwidth requirements of small and medium-sized clusters, the hierarchical log and incremental synchronization schemes in the Raft

algorithm are improved, and the two-phase commit is combined to achieve consistent consensus on the protection strategy.

Logs are categorized into two distinct types based on their urgency and functional importance: high-priority and low-priority logs. High-priority logs comprise critical security-related data, such as malicious IP blacklists and LSTM-based detection metrics, which directly influence attack identification and defense mechanisms and thus necessitate real-time submission and synchronization. In contrast, low-priority logs include non-essential information like traffic statistics and historical node status records, which have limited impact on real-time decision-making and can be transmitted in batches to conserve bandwidth. To mitigate the high overhead associated with full log synchronization, an incremental synchronization strategy is introduced. This approach transmits only newly added log entries rather than the entire log file. For instance, when updates occur – such as the addition of ten new malicious IP hashes or increments in LSTM detection metrics – only the differential data is synchronized. By employing a log entry and delta-differencing algorithm, redundant file transfers are effectively avoided, enhancing synchronization efficiency.

Experiments show that within 0 to 8 minutes, full log synchronization consumes more bandwidth, while incremental log synchronization consumes less. Incremental log synchronization saves more bandwidth than full log synchronization, giving it a significant advantage in bandwidth savings. Incremental log synchronization saves more bandwidth than full log synchronization, giving it a significant advantage in bandwidth savings. To ensure consistency and conflict resolution during incremental log synchronization, this paper introduces a version vector based conflict detection mechanism and checkpoint recovery strategy. Each node maintains a version vector that records the range and hash value of synchronized log entries; during synchronization, the receiver identifies missing or conflicting entries by comparing version vectors. If a log conflict is detected, the "last write wins" strategy is adopted, with the entry with a longer term and updated index as the standard, and the rollback and rewrite of the conflict log are triggered. In addition, nodes regularly persist the submitted log status as checkpoints, load the status from the nearest checkpoint during crash recovery, and then synchronize missing incremental logs from other nodes based on version vectors to ensure final consistency (Figure 6).

The LSTM-generated protection policy, referred to as the "configuration log," requires absolute consistency across all nodes to prevent attack traffic from bypassing vulnerable points. To enforce this, a two-phase policy

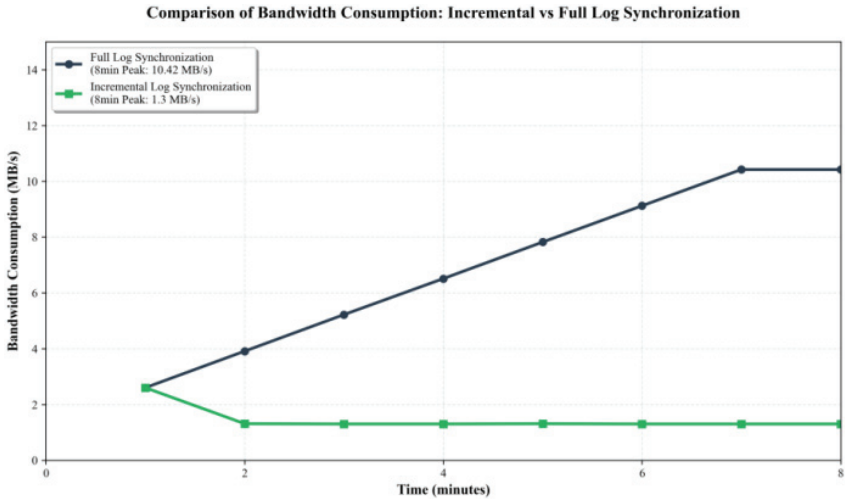

Figure 6 Incremental log synchronization bandwidth comparison.

submission consensus mechanism orchestrates synchronized deployment. In the preparation phase, the leader node broadcasts a "prepare" request containing the defense policy metadata – including expiration timestamp and version number – to all healthy nodes. Recipient nodes validate the policy's integrity; successful verification triggers resource locking and a "ready" acknowledgment, while failures return structured error codes. During the commit phase, the leader evaluates responses: upon receiving "ready" from a majority quorum, it issues an "execute" command with the original expiration timestamp, prompting nodes to activate the policy post-deadline and confirm completion. If consensus fails, the leader triggers a rollback via "cancel" commands, forcing nodes to maintain resource locks and preserve state consistency. Experimental data demonstrates that while both approaches exhibit escalating latency with cluster scaling, the two-phase commit mechanism consistently outperforms traditional single-phase synchronization – reducing delays by 38–62% across node counts from 10 to 200 (Figure 7). This efficiency stems from its incremental validation model, which minimizes redundant communication rounds compared to monolithic policy pushes.

In summary, the improved Raft algorithm achieves efficient and consistent consensus on protection strategies while meeting bandwidth constraints through hierarchical logging, incremental synchronization, and two-phase commit.

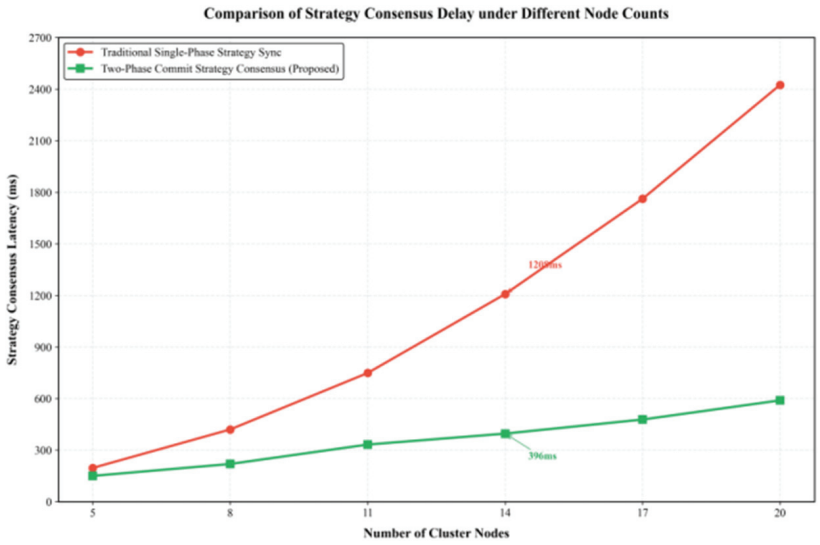

Figure 7 Comparison of consensus delays between single-phase and two-phase commit strategies at different numbers of nodes.

4 Experimental Verification and Result Analysis

4.1 Experimental Environment Settings

In order to verify the progressiveness of the proposed method, experimental research was carried out. This time, 10 Huawei FusionServer Pro 2288H V5 servers were used to form a distributed cluster. Each server was equipped with 2 Intel Xeon Platinum 8260 CPUs with 24 cores and 48 threads, 256 GB DDR4 memory, and NVMe SSD arrays for storage. The read and write IOPS were not less than 500k. All servers were interconnected through gigabit Ethernet switches, and Mininet and NS-3 were used to jointly build a network simulation environment to simulate the real topology of a wide area network. The network topology was partially mesh connected, with a basic delay between nodes set at 50 ms $\pm$ 10 ms, jitter of 15 ms, bandwidth limit of 100 Mbps, and packet loss rate set at 0.5%.

The underlying platform of blockchain adopts Hyperledger Fabric v2.2, deploying 3 sorting nodes and 5 endorsement nodes, and replacing the consensus module with the improved Raft algorithm implemented in this article. There is a total of 50 consensus nodes in the node scale, including simulated malicious nodes: 4 Byzantine nodes are set in the regular scenario, and 8 nodes are expanded in the attack scenario. The Byzantine node behavior

pattern is configured with a 30% probability of launching a double voting attack, a 30% probability of selectively rejecting forwarded log entries, and a 40% probability of randomly delaying responses. In privacy attack simulation, 20% of nodes simulate traffic analysis attacks and attempt to trace pseudonyms through time correlation analysis; 15% of nodes simulate denial of service attacks and send a large number of invalid requests to leaders; 10% of nodes simulate Byzantine behavior and broadcast conflict messages.

This experiment uses two types of data sources – the publicly available Pecan Street energy dataset, which includes electricity consumption data of 1000 households at 15 minute intervals and actual operating data of a certain wind farm, including power generation data at 1 minute intervals. After desensitization, 5000 normal transaction records were generated for routine scenario testing. Inject an additional 15000 log records containing abnormal behavior in the attack scenario, with abnormal patterns including sudden high-frequency requests, exceeding the normal threshold by 3 times, and periodic rejection of responses. The data is randomly divided into a training set and a testing set in a 7:3 ratio.

The training data required for the LSTM health assessment model comes from the node status logs collected during the above running process, covering temporal data in four dimensions: consensus performance, resource load, traffic anomalies, and historical behavior, with a time span of 30 consecutive days. Tags are generated through post analysis: if a node triggers a consensus timeout, view change, or throughput drop of more than 50% during a certain time period, all time steps within that time period are marked as unhealthy (0), and the rest are marked as healthy (1). To ensure the quality of the labels, three domain experts independently annotated and obtained a majority consensus result. The model is updated once a week, using newly added logs and historical data to merge and retrain to adapt to the evolution of node behavior.

In addition, the training parameters for the LSTM health assessment model are as follows: input time step of 50, feature dimension of 4, including consensus performance, resource load, traffic anomalies, and historical behavior. The model adopts a double-layer LSTM structure, with 64 units in the first layer and 32 units in the second layer. The Dropout rate is set to 0.2 to prevent overfitting. The output layer is mapped to a health score after being activated by Sigmoid. The training uses the Adam optimizer with a learning rate of 0.001, batch size of 32, 100 rounds of training, and adopts an early stopping mechanism. When the validation loss does not decrease for 10 consecutive rounds, the training is terminated.

During the testing process, each experiment was repeated 30 times, and the average was taken as the final result. At the same time, the standard deviation was calculated to evaluate the stability of the results. Based on this, carry out subsequent experimental research.

4.2 Consensus Performance Comparison

This study evaluates the performance of three consensus mechanisms: traditional Raft, Raft enhanced with a pre-voting mechanism, and Raft further integrated with LSTM-based health detection and pre-voting. The experiment simulates the operation of distributed nodes to generate sample data. A two-layer LSTM model is trained to assess node health status. Each model configuration undergoes 50 rounds of testing under both normal conditions (involving two abnormal nodes and 5000 logs) and attack scenarios (with four abnormal nodes and 15,000 logs). Consensus performance is measured using the average consensus latency. The experimental results are illustrated in Figure 8.

Experimental results show that the LSTM health awareness mechanism achieves the lowest consensus latency in both scenarios and is more resistant to latency fluctuations in attack scenarios.

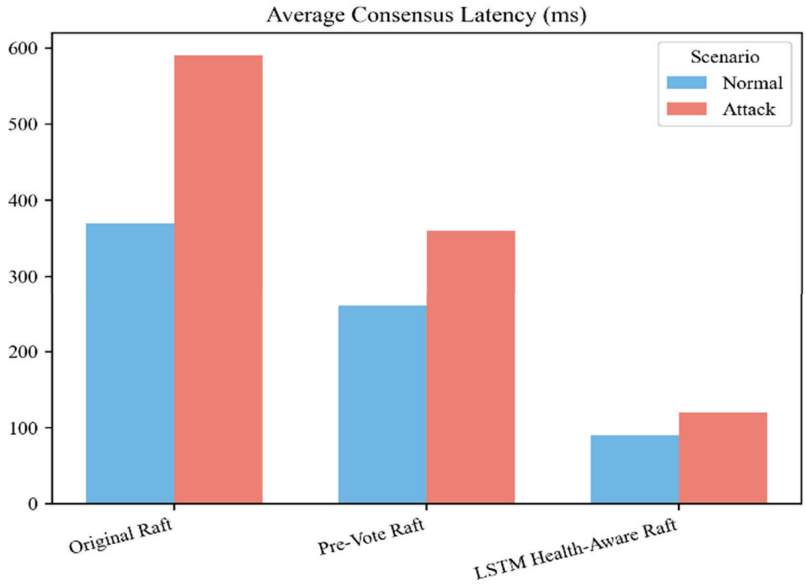

Figure 8 Comparison of consensus delays.

4.3 Experimental Comparison of Privacy-preserving Algorithms

To verify the performance of a dynamic pseudonym-based lightweight zK-SNARK scheme in supporting on-chain energy data transactions, this section conducts comparative experiments on the dynamic pseudonym-based lightweight zK-SNARK scheme (the proposed scheme) with no pseudonym, static pseudonym, and traditional zK-SNARK schemes. The following metrics are used to experimentally verify and compare the four schemes from the perspectives of performance, privacy, and usability.

Performance dimensions

TPS (transactions per second): The number of on-chain transactions successfully processed per second, reflecting the solution's high-frequency trading processing capabilities.

Average response time: The average time it takes for a transaction to be initiated and packaged into a block, reflecting the user experience.

Privacy dimension

Identity association rate: This indicator is calculated through simulated link attacks, where the attacker has some background knowledge, such as timestamps and amount ranges of several transactions of a certain user, and attempts to match fake transactions on the chain with their real identity. In the experiment, each scheme was subjected to 1000 link attacks, and the percentage of successful attacks to the total number of attempts was calculated as the identity association rate. To simulate real-world threats, the attack model not only considers single transaction associations, but also incorporates multiple rounds of transaction associations – attackers can use the appearance patterns of the same pseudonym in different blocks to analyze and infer user behavior patterns. At the same time, background knowledge attacks are also taken into account: attackers may obtain external information such as electricity usage habits and trading hours of some users in order to narrow down the matching scope.

Availability dimension

Node fault holdover rate: After a faulty node goes offline, the system's remaining performance accounts for the ratio of its pre-fault performance to its fault tolerance.

The TPS of the proposed scheme is 6 times that of traditional zk-SNARKs schemes and 4.43 times that of non-pseudonymous schemes (Figure 9);

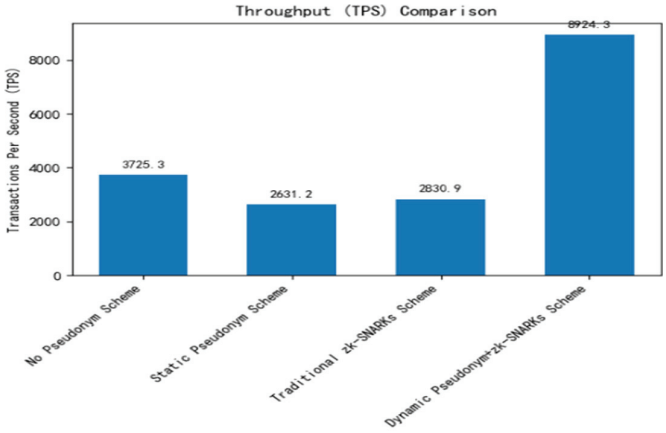

Figure 9 TPS comparison chart.

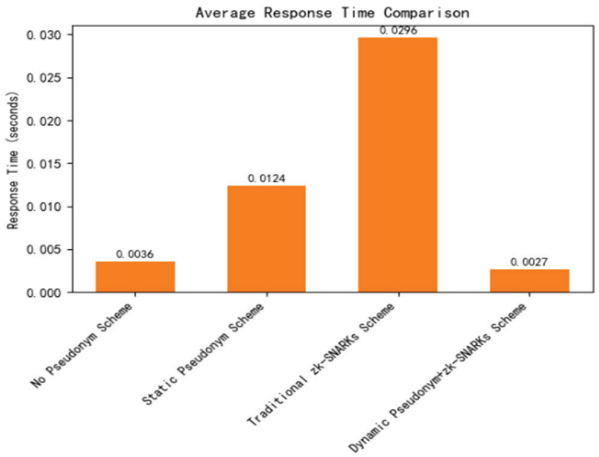

Figure 10 Average response time comparison.

the average response time is only 2.76% of that of traditional zk-SNARK schemes (Figure 10). The key reason lies in the batch proof mechanism, which combines the proof and verification of multiple transactions into a single one, thereby reducing on-chain computation and storage costs while matching block size, thus avoiding queuing of transactions outside the block.

The proposed scheme in this paper is significantly lower than the other three schemes. The identity correlation rate refers to the probability of associating on-chain data with the real identity of the user Phoxinus phoxinus

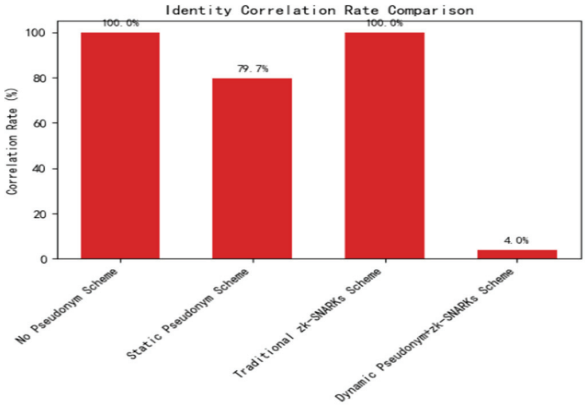

Figure 11 Identity association rate comparison chart.

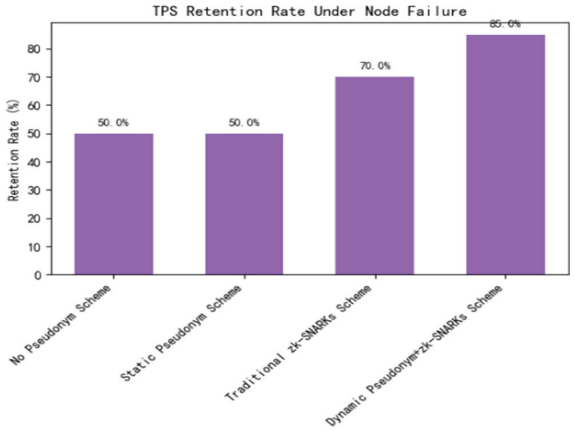

Figure 12 Node failure holdover rate.

subsp. phoxinus, indicating that the improved zero-knowledge proof algorithm based on dynamic pseudonyms proposed in this paper exhibits high privacy (Figure 12).

Figure 13 demonstrates the superiority of our proposed solution over both the distributed node approach and the conventional zk-SNARK scheme under pseudonym-free and static pseudonym configurations. This advantage is attributed to our design's ability to effectively distribute the transaction load across the network; when a node fails, the remaining nodes seamlessly take over the generation of batch proofs, preventing any significant degradation in system performance. By contrast, in traditional zero-knowledge proof setups,

where each node is individually responsible for generating a single proof, the failure of even one node leads to a marked decline in overall performance.

From the experimental comparison results of the pseudonym-free scheme, static pseudonym scheme, traditional zk-SNARK scheme and the proposed scheme, which is a lightweight zk-SNARK scheme based on dynamic pseudonyms, it can be seen that this scheme can provide a high degree of privacy protection while providing a high on-chain transaction processing speed and availability. It improves the defects of the pseudonym-free scheme's lack of privacy protection, the static scheme's identity association, and the poor performance of the traditional zk-SNARKs scheme and is suitable for on-chain sensitive data transaction scenarios.

4.4 Feasibility Analysis of Deployment

To evaluate the feasibility of the proposed solution in practical energy trading scenarios, this section conducts a comprehensive analysis from three dimensions, computational resource requirements, communication overhead, and system compatibility, and compares it with existing energy blockchain projects.

4.4.1 Analysis of computing resource requirements

The computational cost of this scheme mainly comes from three modules: dynamic pseudonym generation, zk-SNARK batch proof generation and verification, and LSTM health assessment inference. In the experimental environment, the resource consumption of each module is shown in Table 1.

From Table 1 it can be seen that the average computational cost (pseudonym generation) of a single transaction is extremely low, and the main

Table 1 Calculation resource consumption of each module

Module	CPU Usage (Cores)	Memory Usage (MB)	Time Per Operation (ms)	Applicable Scene
Dynamic pseudonym generation	0.2	8	0.3	Per transaction
Zerod-SNARKs proof generation	2.5	256	185	Per block (100 transactions)
zk-SNARKs proof verification	0.8	64	42	Per block
LSTM health assessment	0.3	128	2.3	Every election cycle

Table 2 Communication cost analysis (single block, 100 transactions)

Communication Type	Data Volume (KB)	Frequency	Daily Average Traffic (MB)	Proportion (%)
Trading broadcast	256	Every transaction	221	18.5
Consensus message	48	Each block	41	3.4
Incremental log synchronization	128	Each block	111	9.3
Batch proof	32	Each block	28	2.3
Total	464	–	401	100

computationally intensive operation (proof generation) is executed in batches by block, significantly reducing the computational cost per unit transaction. Taking the high-frequency scenario of 100 transactions per second as an example, the node requires approximately 4 cores of CPU resources and 450 MB of memory, which is within the current mainstream industrial server capacity. Compared with existing solutions, the energy trading solution based on Ethereum can consume more than 8 cores of CPU per node due to gas limitations and PoW consensus; the Hyperledger Fabric solution occupies approximately 800 MB of memory under the same transaction load. This solution has significant advantages in resource efficiency and is suitable for deployment in consortium chain node server environments.

4.4.2 Evaluation of communication costs

The communication overhead mainly includes four parts: transaction broadcasting, consensus messages, log synchronization, and proof transmission. At a scale of 50 nodes, the measured traffic of various types is shown in Table 2.

According to Table 2, its daily communication volume is about 400 MB. Calculated based on a bandwidth of 100 Mbps, it occupies about 0.37 Mbps of bandwidth, which is much lower than the typical enterprise bandwidth limit. Compared with full log synchronization, incremental log synchronization saves about 68% of bandwidth and effectively reduces network pressure. In the wide area network deployment scenario (with a delay of 50 ms), the end-to-end confirmation time for a single transaction is about 1.2 s, meeting the second level trading demand of the electricity spot market.

4.4.3 Compatibility analysis of existing systems

To quantitatively evaluate the compatibility of the proposed solution with existing systems, three typical energy systems were selected for integration testing: smart meter acquisition system (based on MQTT), park energy

Table 3 Compatibility test results with existing systems

Integrated Object	Protocol Type	Success Rate of Docking (%)	Average Delay (ms)	Throughput Effect (%)	Data Format Conversion Time (ms)
Smart meter collection system	MQTT	100	12.3	+2.1	0.8
Park energy management system	RESTful API	99.8	28.7	−1.5	1.2
Power dispatch data platform	JDBC	100	45.2	−3.4	2.5

management system (based on RESTful API), and power dispatch data center (based on JDBC). The test results are shown in Table 3.

An analysis of Table 3 shows that the integration with the smart meter collection system performs the best, thanks to the MQTT lightweight protocol and the built-in protocol adaptation layer of the solution, with an average delay of only 12.3 ms and almost no negative impact on the original system throughput. The success rate of integration with the park's energy management system is 99.8%, with a few failures caused by current limiting of the original system interface. After retries, all attempts were successful, and the delay mainly comes from the serialization of RESTful requests and network transmission. The integration with the power dispatch data center involves complex SQL conversion, with slightly higher latency, but still meets offline synchronization requirements. The data format conversion time is within 3 ms, which proves that the data adaptation layer of the scheme has high efficiency. Overall, the proposed solution has good compatibility with existing energy systems, controllable integration costs, and practical deployment conditions.

5 Conclusion

This paper presents an integrated solution that addresses three critical technical challenges faced by conventional blockchain approaches in energy data transactions. Specifically, the dynamic pseudonym mechanism effectively mitigates identity-based transaction correlation risks, while the lightweight zk-SNARK batch proof technique significantly enhances transaction processing speed. Furthermore, the enhanced Raft consensus algorithm

ensures robust consistency and system stability under dynamic conditions. Experimental validation confirms that the proposed solution outperforms traditional methods across key metrics, with the optimized Raft algorithm demonstrating superior scalability and fault tolerance. By achieving an optimal equilibrium between privacy preservation and operational efficiency, this framework not only caters to the unique demands of energy data trading but also fosters cross-disciplinary innovation by bridging privacy-centric computation with distributed consensus methodologies. These advancements underscore its practical applicability and broader value in real-world implementations.

References

[1] W. Zhang, Z. Guo, G. Han, H. Zhu and Q. Liu, "DPSP: A Dynamic Pseudonym Swap Program Based Location Privacy Protection Algorithm for Internet of Vehicles", IEEE Transactions on Network Science and Engineering, vol. 11, no. 5, pp. 4525–4535, Sept.–Oct.2024, doi: 10.1109/TNSE.2024.3392709.

[2] Y. Li, Y. Yin, X. Chen, J. Wan, G. Jia and K. Sha, "A Secure Dynamic Mix Zone Pseudonym Changing Scheme Based on Traffic Context Prediction," in IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 7, pp. 9492–9505, July 2022, doi: 10.1109/TITS.2021.3125744.

[3] JinRu Wang. Wireless key generation algorithm for RFID system based on partial pseudonym ID[J]. Computer Engineering and Applications, 2018, 54(01): 128–132.

[4] H. Qi, Y. Cheng, M. Xu, D. Yu, H. Wang and W. Lyu, "Split: A Hash-Based Memory Optimization Method for Zero-Knowledge Succinct Non-Interactive Argument of Knowledge (zk-SNARK)," in IEEE Transactions on Computers, vol. 72, no. 7, pp. 1857–1870, 1 July 2023, doi: 10.1109/TC.2023.3235975.

[5] D. A. Luong and J. H. Park, "Privacy-Preserving Blockchain-Based Healthcare System for IoT Devices Using zk-SNARK," in IEEE Access, vol. 10, pp. 55739–55752, 2022, doi: 10.1109/ACCESS.2022.3177211.

[6] Z. Guan, Z. Wan, Y. Yang, Y. Zhou and B. Huang, "BlockMaze: An Efficient Privacy-Preserving Account-Model Blockchain Based on zk-SNARKs," in IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 3, pp. 1446–1463, 1 May-June 2022, doi: 10.1109/TDSC.2020.3025129.

[7] D. A. Luong and J. H. Park, "Privacy-Preserving Identity Management System on Blockchain Using Zk-SNARK," in IEEE Access, vol. 11, pp. 1840–1853, 2023, doi: 10.1109/ACCESS.2022.3233828.

[8] HaoNan Wang, JingYu Wang, LiXin Liu, et al. Cross-chain transaction scheme based on multi-signature and zero-knowledge proof[J]. Computer Engineering and Design, 2025, 46(06): 1694–1702. doi: 10.16208/j.issn1000-7024.2025.06.021.

[9] YiCong Li, KuanJiu Zhou, ZiZhong Wang. Research on blockchain privacy protection based on zero-knowledge proof[J]. Space Control Technology and Applications, 2022, 48(01): 44–52.

[10] Ting Wu, ShaSha Zhang. Scalable privacy protection scheme for blockchain based on homomorphic encryption and zero-knowledge proof[J]. Journal of Computer Application Research, 2025, 42(07): 1939–1947. doi: 10.19734/j.issn.1001-3695.2025.01.0002.

[11] Miao Jia, ZhongYuan Yao, WeiHua Zhu, et al. Progress and prospects of zero-knowledge proof-enabled blockchain[J]. Computer Applications, 2024, 44(12): 3669–3677.

[12] Kai Zhou, Fu Chen, TianYuan Lu, et al. A review of blockchain consensus algorithms [J/OL]. Computer Science, 1–25 [2025-09-18]. https://link.cnki.net/urlid/50.1075.TP.20250317.1848.036.

[13] Yu, X. Deng, W. Xiong and W. Li, "CB-Raft: A Byzantine Fault-Tolerant Raft Consensus Mechanism Based on Comprehensive Evaluation Partitioning," 2024 2nd International Conference On Mobile Internet, Cloud Computing and Information Security (MICCIS), Changsha City, China, 2024, pp. 34–38, doi: 10.1109/MICCIS63508.2024.00013.

[14] ShuZhi Li, YiJie Zou, XiaoHong Deng, et al. RB-Raft: A Raft consensus algorithm with Byzantine resistance[J]. Journal of Computer Application Research, 2022, 39(09): 2591–2596. doi: 10.19734/j.issn.1001-3695.2022.03.0090.

[15] Li Li, HaoZe Li, Tao Li . Multi-master Byzantine fault-tolerant consensus mechanism based on Raft[J]. Journal of Guangxi Normal University (Natural Science Edition), 2024, 42(03): 121–130. doi: 10.16088/j.issn.1001-6600.2023100805.

[16] A. Fitwi, Y. Chen and S. Zhu, "A Lightweight Blockchain-Based Privacy Protection for Smart Surveillance at the Edge," 2019 IEEE International Conference on Blockchain (Blockchain), Atlanta, GA, USA, 2019, pp. 552–555, doi: 10.1109/Blockchain.2019.00080.

[17] MingQuan Zhang, XinyYu Cao . Improved Raft consensus algorithm based on alliance chain microgrid transactions[J]. Computer Application

Research, 2024, 41(10): 2911–2917. doi: 10.19734/j.issn.1001-3695.2024.01.0010.

[18] Xing Liang . Research on privacy protection of trajectory data based on generative adversarial network[D]. Chang'an University, 2024. doi: 10.26976/d.cnki.gchau.2024.001035.

Biographies

Rui Xin, male, is master's degree candidate and a senior engineer. His main research focuses on information system construction, network security, and data management.

ShaoYing Wang, female is a master's degree candidate and intermediate engineer. Her main research areas include data management and big data analytics.

Xin Lu, female, is a master's degree candidate, and senior engineer. Her main research areas include information system construction and data management.

YanYan Lu, female, is a master's degree candidate and engineer. Her main research directions are big data analysis and applications.

LiPing Yang, female, is a master's degree candidate and intermediate engineer. Her main research areas include data management and big data analytics.