International Journal of Electric Power and Energy Studies
ISSN: 3006-2004 (Print), ISSN: 3006-0826 (Online) | Volume 5, Number 1, Year 2026
DOI: https://doi.org/10.62051/ijepes.v5n1.04
Journal homepage: https://wepub.org/index.php/IJEPES/index

A Lightweight Aggregation Scheme for Multidimensional Charging Privacy Data based on Consortium Blockchain

Guilan Wang[1,2], Jian Cui[1], Chunhong Duo[1,3,*]

[1] College of Control and Computer Engineering, North China Electric Power University, Baoding 071003, P.R. China

[2] Hebei Key Laboratory of Knowledge Computing for Energy & Power, Baoding 071003, P.R. China

[3] Engineering Research Center of Intelligent Computing for Complex Energy Systems, Ministry of Education, Baoding 071003, P.R. China

[*]Corresponding Author: Chunhong Duo

ABSTRACT

With the growing integration of charging piles into the smart grid, efficiently aggregating privacy-containing charging data, dynamically optimizing charging strategies, and enhancing charging efficiency have become key future development directions. Current data aggregation schemes, mostly based on homomorphic encryption, place high computational demands on charging piles and may lead to centralized data management issues. To address these challenges, a lightweight data aggregation scheme based on consortium blockchain is proposed. Firstly, non-interactive symmetric encryption and aggregate signatures are used to reduce computational and communication overhead. Symmetric encryption ensures efficient encryption and decryption processes, while aggregate signatures compress multiple signatures into one, reducing storage and verification costs. Secondly, a hierarchical distributed data aggregation model is designed to achieve decentralization, distributing aggregation tasks across multiple layers to enhance system scalability and robustness. Thirdly, a dual-layer consensus algorithm is proposed based on the architecture of charging piles and edge cloud servers. This algorithm ensures low latency and system robustness at the charging pile layer, while the edge cloud server layer can resist Byzantine attacks. By balancing efficiency and security, this approach optimizes resource utilization and enhances privacy protection. Finally, experiments demonstrate that the proposed scheme significantly reduces computational and communication overhead and improves efficiency.

KEYWORDS

Consortium Blockchain; Multidimensional Data Aggregation; Consensus Algorithm; Privacy Protection; Lightweight.

1. INTRODUCTION

Facing the new pattern of future energy supply and demand, the State Council of China issued the "2024-2025 Energy Conservation and Carbon Reduction Action Plan", implementing supportive policies such as facilitating the passage of new energy vehicles. The rapid development of electric vehicles has driven the widespread construction of charging infrastructure [1]. As of October 2024, the total number of charging infrastructure in China has exceeded 10 million units [2]. Charging piles, as part of the charging infrastructure, can collect key charging data. Through big data technology, the analysis of key data can assist in grid expansion and orderly charging control for electric vehicles [3].

Charging piles collect multidimensional data such as current, voltage, temperature, and charging time [4]. However, these charging data may involve user privacy information, posing security risks such as privacy leakage. With the increasing deployment of charging piles, a large amount of charging data will be generated. Direct collection and storage of charging data by grid companies may face centralized issues such as single-point failures. Data aggregation technology [5] involves collecting and aggregating data from multiple data sources. In the context of charging pile data aggregation, it is essential not only to achieve data aggregation but also to ensure user privacy security. Charging piles have limited computational performance and storage capacity, and grid companies need to process large amounts of charging data while handling other business operations. Therefore, the solution must achieve efficient and trustworthy data aggregation while ensuring privacy protection and reducing the computational and storage overhead on both the charging pile and grid company sides.

2. RELATED WORK

Current research on privacy-preserving data aggregation can be broadly categorized into three types. The first type is based on homomorphic encryption. Chen et al. [6] used the Okamoto-Uchiyama encryption algorithm with homomorphic properties to protect user privacy. Azhar et al. [7] employed Paillier homomorphic encryption to encrypt user data and perform aggregation on the encrypted data. Hu et al. [8] improved the Boneh-Goh-Nissim (BGN) public-key encryption scheme with a blinding factor to achieve privacy protection and data aggregation and management. However, these homomorphic encryption-based schemes generate a large amount of redundant data, increasing computational, communication, and storage overhead for charging piles, which have limited computational resources.

To reduce computational and communication overhead, the second type of scheme is based on masking encryption. Li et al. [9] used elliptic curve-based data masking technology to encrypt data for privacy protection. Xue et al. [10] proposed a masking-based data aggregation scheme in the absence of a trusted authority, reducing the computational overhead on smart meters. Su et al. [11] proposed a scalable and decentralized data aggregation scheme for edge IoT nodes, using masking-assisted encryption during the data collection phase to achieve privacy protection. However, these schemes are vulnerable to man-in-the-middle attacks, which can expose aggregation results. Additionally, since the masks must sum to zero, if a charging pile node fails to submit data on time, it will affect the final aggregation result, resulting in poor robustness.

As a supplement to the first two types of schemes, the third type is based on secret sharing algorithms, often combined with the first two types of schemes. Zhang et al. [12] proposed a multidimensional data aggregation scheme based on threshold-variable secret sharing to address the high energy consumption issue in wireless sensor networks. Mustafa et al. [13] proposed a smart meter data aggregation method based on Shamir's secret sharing. Chen et al. [14] studied a persistent computation mechanism based on multi-secret sharing, combined with masking technology to convert secrets into protected data. However, these schemes do not consider the risk of data tampering or loss, and the need to share multiple secret data increases communication overhead. Moreover, the decryptor must be online simultaneously to obtain the complete data.

The aforementioned schemes achieve data aggregation while considering privacy protection but still have some shortcomings. For example, they only consider one-dimensional data [6-10]. Homomorphic encryption [6-8] imposes high computational resource requirements on charging piles. Charging piles may experience failures, exits, or joins, requiring consideration of the scheme's robustness [12-14]. Additionally, most schemes are centralized [9-11], posing single-point failure security risks. Blockchain technology, with its inherent properties of decentralization, traceability, and tamper resistance, has increasingly been applied in the energy sector. Li et al. [15] proposed a trusted data aggregation scheme based on a double-layer blockchain, using blockchain to achieve

autonomous charging data aggregation. Loukil et al. [16] proposed a privacy-preserving IoT data aggregation scheme based on blockchain and homomorphic encryption, achieving data aggregation through smart contracts. Shafeeq et al. [17] proposed a blockchain-based data aggregation scheme that can detect end-to-end integrity of aggregation results and identify malicious aggregators, while using sidechains to improve system efficiency and scalability. Since signature algorithms are needed to verify integrity during data aggregation, algorithms supporting batch signature verification can reduce the time overhead of signature verification. Reference [7] uses the ECDSA signature algorithm, which does not support batch signature verification, while references [6,18] use bilinear mapping signature algorithms that support batch signature verification, but the bilinear pairing time overhead during signature verification is time-consuming compared to other operations.

To address the above issues, this paper proposes a lightweight multidimensional charging privacy data aggregation scheme based on consortium blockchain. The main contributions of this paper are as follows:

To address the limited computational resources of charging piles and centralized data storage, this paper designs a lightweight data aggregation scheme based on consortium blockchain and non-interactive symmetric encryption, proposing a hierarchical distributed data aggregation model to effectively reduce computational and storage overhead.

Based on the dual-layer structure of charging piles and edge cloud servers in the aggregation model, a dual-layer consensus algorithm is proposed to complete consensus tasks with low latency at the charging pile layer while ensuring robustness; the edge cloud server layer can resist Byzantine attacks.

Experimental tests and analyses are conducted on the proposed scheme, comparing computational overhead, communication overhead, and consensus efficiency. The comparison results demonstrate that the proposed scheme can effectively reduce computational and communication overhead and improve data aggregation efficiency.

3. PRELIMINARIES

3.1. Aggregate Signature Scheme

This paper uses an aggregate signature scheme [8] to verify data integrity. Based on the security parameter λ, a cyclic group G of order q is defined, with generator P. A secure collision-resistant hash function is randomly selected: $H: \{0,1\}^* \rightarrow Z_q^*$. The signer randomly selects $x_{ID_i} \in Z_q^*$ as the private key and computes the public key $P_{ID_i} = P^{x_{ID_i}}$.

During signing, a random $u_i \in Z_q^*$ is selected, and $U_i = P^{u_i}$ is computed. Then, the signature $s_i = u_i + H(m_i)x_{ID_i}$ is computed, where m_i is the message to be signed. The tuple $(m_i||U_i||s_i)$ is sent to the verifier.

The verifier verifies the correctness of a single signature using Equation (1):

$$P^{s_i} = U_i P_{ID_i}^{H(m_i)} \tag{1}$$

The correctness of n signatures is verified using Equation (2):

$$P^{\Sigma_{i=1}^{n} s_i} = \prod_{i=1}^{n} U_i P_{ID_i}^{H(m_i)} \tag{2}$$

3.2. Non-Interactive Symmetric Encryption

A symmetric bivariate polynomial $F(x,y) = [x^0, x^1, \ldots, x^{t-1}]D[y^0, y^1, \ldots, y^{t-1}]^T$ is constructed using a t-order symmetric matrix D. Each user U_i has a public identity $x_i \in Z_q^*$. The trusted authority computes $S_i(y) = F(x_i, y)$ and sends it to U_i through a secure channel.

When user U_i sends a message to user U_j, U_i first computes $K_{ij} = S_i(x_j) = F(x_i, x_j)$ without the help of U_j. Similarly, U_j can compute $K_{ji} = S_j(x_i) = F(x_j, x_i)$. Since $K_{ij} = F(x_i, x_j) = F(x_j, x_i) = K_{ji}$, non-interactive symmetric key distribution is achieved [18].

3.3. Blockchain and Consensus Algorithms

In consortium blockchains, read and write permissions or node joining are determined by the consortium nodes. It is suitable for a consortium or industry and is used in organizations that require mutual communication [19]. Therefore, this paper uses a consortium blockchain to manage nodes. Common consensus algorithms in consortium blockchains include PBFT, MBFT [19], and Raft [20]. To improve consensus efficiency, Yuan et al. [21] proposed the DL_RBFT dual-layer consensus algorithm, which reduces consensus latency and improves throughput by grouping and introducing supervision and reputation mechanisms.

4. SYSTEM MODEL

To reduce the computational and storage overhead on both the charging pile and grid company sides, an edge cloud server layer is introduced. Charging piles and edge cloud servers form a dual-layer blockchain network. Charging piles, as lightweight nodes, are responsible for data aggregation within their regions, reducing computational overhead. The edge cloud server layer, as the intermediate layer, handles global aggregation and stores aggregated data. It also serves as full nodes in the blockchain, ensuring data consistency through consensus algorithms and preventing malicious cloud servers from tampering with aggregated data. The system model achieves the design goals of data integrity verification, privacy protection, robustness, lightweight, and decentralization.

The system model is divided into three layers: the charging pile layer, the edge cloud server layer, and the data request layer. As shown in Figure 1, it includes four entities: charging nodes (CNs), edge cloud server nodes (ESs), the power company (PC), and the trusted authority (TA).

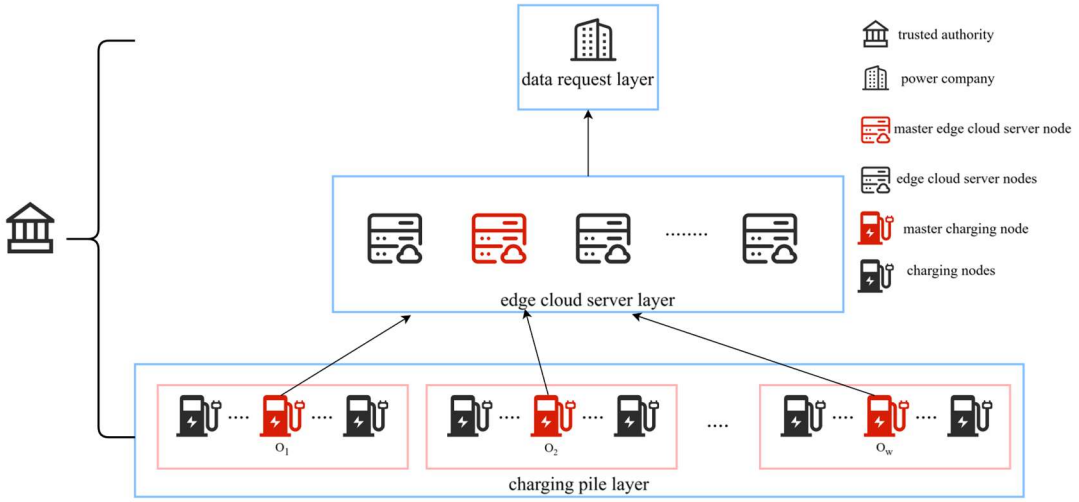


Figure 1. System model

5. SYSTEM SCHEME

This section presents the specific process of the aggregation scheme, including five parts: system initialization, data aggregation, data decryption, dynamic node management, and the dual-layer consensus algorithm. For ease of understanding, Table 1 lists the symbols and descriptions used in this paper.

Table 1. Symbols And Descriptions

Symbols	Descriptions
H_1	secure collision-resistant hash function
$F(x,y)$	symmetric bivariate polynomial
$a_1, a_2, \ldots, a_l$	l large positive integers
ES_i	i-th ES
CN_{ij}	j-th CN in region O_i
x_{ID}， S(y)	private key, Secret Polynomial
d_{max}	Maximum value of the data vector
$\vec{m}_{ij}$	CN_{ij} collects multidimensional data
ID_x	ID of PC、ES_i、CN_{ij}

5.1. Initialization

Sub-headings should be typeset in boldface italic and capitalize the first letter of the first word only. Section numbers to be in boldface roman.

Assume there are m ESs in the system, where ES_a represents the master edge cloud server node, and ES_i represents ordinary edge cloud server nodes. Each region O_i has an elected master charging node CN_{ia}, and region O_i initially contains s_i CNs: $\{CN_{i1}, CN_{i2}, \ldots, CN_{is_i}\} \triangleq U_i$, where CN_{ij} represents the j-th CN in region O_i. Thus, we have: $\sum_{i=1}^{w} s_i = s$.

The TA first generates system parameters, assists other entities in registering with the system, and distributes key generation polynomials and keys through secure channels, as shown in Figure 2.

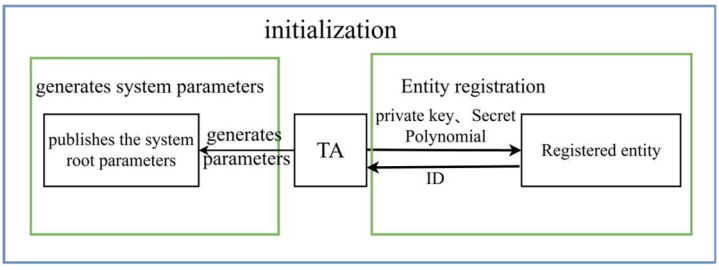


Figure 2. initialization process

Based on the security parameter λ, the TA defines a cyclic group G and constructs a symmetric bivariate polynomial $F(x,y)$. Assuming the charging data is l-dimensional, the TA randomly selects l large positive integers $a_1, a_2, \ldots, a_l$ that are pairwise prime, satisfying $a_i \gg sd_{max}, 0 \leq i \leq l$,

where d_{max} is the maximum value of all components of the data vector, and s is the total number of CNs. The TA then calculates the following values using Equation (3):

$$\begin{cases} \hat{A} = a_1 a_2, \ldots, a_l \\ A_i = \hat{A} / a_i \qquad i = 1, 2, \ldots, l \\ \hat{A}_i \equiv A_i \left(A_i^{-1} moda_i \right) \end{cases} \tag{3}$$

The TA randomly selects $H_1 : \{0,1\}^* \rightarrow Z_q^*$ and a public symmetric encryption algorithm Enc. The TA also randomly selects $x \in Z_q^*$ as the system master key, computes the system public key $P_{pub} = P^x$, and publishes the system root parameters $\{q, \mathrm{G}, \mathrm{P}, l, d_{max}, t, \{\hat{A}\}_{i=1}^{l}, \hat{A}, H_1, Enc, P_{pub}\}$.

For entity registration, taking CN as an example, after receiving ID_{ij} from CN_{ij}, the TA computes $S_{ij}(y) = F(h_{ij}, y)$, where $h_{ij} = H_1(ID_{ij})$. The TA then randomly selects $x_{ID_{ij}} \in Z_q^*$, computes the public key $P_{ID_{ij}} = P^{x_{ID_{ij}}}$, and sends $\{x_{ID_{ij}}, S_{ij}(y)\}$ to CN_{ij} through a secure channel, while publishing $\{h_{ij}, P_{ID_{ij}}\}$. Similarly, the TA sends $\{x_{ID_c}, S_c(y)\}$ to the PC and publishes $\{h_c, P_{ID_c}\}$; it also sends $\{x_{ID_i}, S_i(y)\}$ to ES_i and publishes $\{h_i, P_{ID_i}\}$.

5.2. Data Aggregation

Data aggregation is divided into local aggregation and global aggregation, as shown in Figure 3.

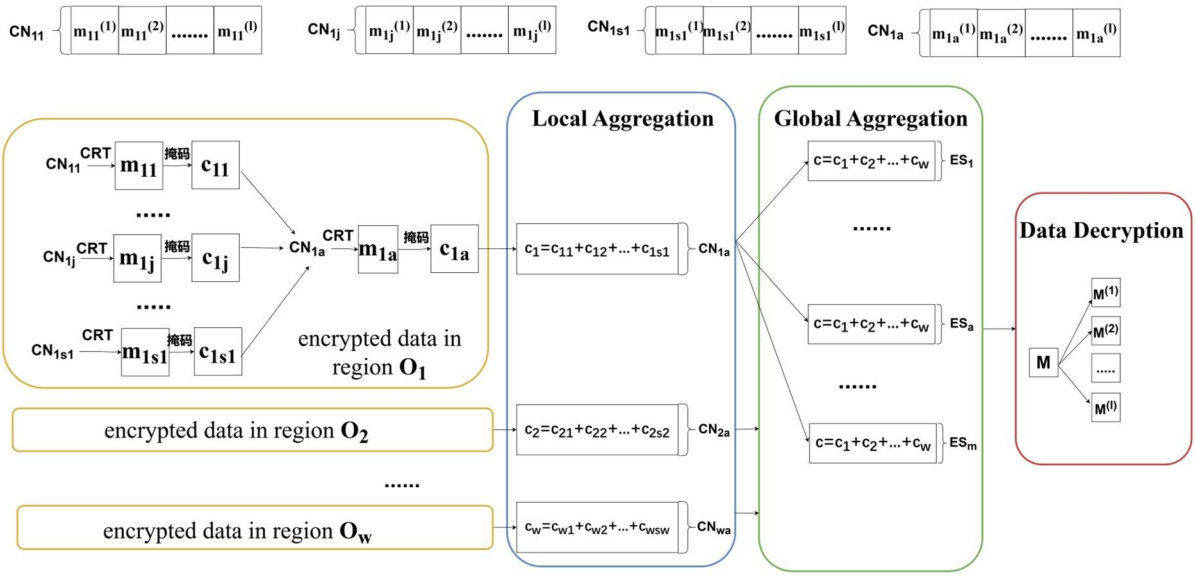


Figure 3. data aggregation process

5.2.1. Local Aggregation

Taking CN_{1j} in region O_1 as an example, CN_{1j} periodically (e.g., every 15 minutes) collects multidimensional data $\vec{m}_{1j} = (m_{1j}^{(1)}, m_{1j}^{(2)}, \ldots, m_{1j}^{(l)})$. It then performs the following steps to encrypt $\vec{m}_{1j}$ and sends the encrypted data to CN_{1a}.

Step 1:CN_{1j} converts $\vec{m}_{1j}$ into a large integer m_{1j} using the Chinese Remainder Theorem (CRT): $m_{1j} = \Sigma_{l=1}^{l} m_{1j}^{(t)} \hat{A}_t = m_{1j}^{(1)} \hat{A}_1 + m_{1j}^{(2)} \hat{A}_2 + \cdots + m_{1j}^{(l)} \hat{A}_l$.

Step 2:For other CN_{1t}, CN_{1j} computes $S_{1j}(h_{1t})$ and derives the local mask $z_{1j} = \Sigma_{t=1,t\neq j}^{s_1} sgn(h_{1j} - h_{1t}) S_{1j}(h_{1t})$.

Step 3:CN_{1j} computes $c_{1j} = m_{1j} + z_{1j} + v_{1j}$, where $v_{1j} = S_{1j}(h_c)$, $K_{1j,a} = S_{1j}(h_{1a})$, $K_{1j,a}$ is the shared symmetric key between CN_{1j} and CN_{1a}. The ciphertext $C_{1j,a} = Enc_{K_{1j,a}}(c_{1j})$ is generated.

Step 4:CN_{1j} randomly selects $u_{1j} \in Z_q^*$, computes $U_{1j} = P^{u_{1j}}$, and then computes the signature $s_{1j,a} = u_{1j} + H_1(C_{1j,a}||ID_{1j}||T_{1j,a}) x_{ID_{1j}}$, where $T_{1j,a}$ is the timestamp when CN_{1j} generates $s_{1j,a}$.

Step 5:CN_{1j} sends $Msg_{1j,a} = (C_{1j,a}||ID_{1j}||T_{1j,a}||U_{1j}||s_{1j,a})$ to CN_{1a}.

After receiving $Msg_{1j,a}$, CN_{1a} first checks the validity of the timestamp $T_{1j,a}$. After receiving $Msg_{1j,a}$ from all CN_{1j} in the region, CN_{1a} verifies the correctness of the signatures using Equation (4):

$$P^{\Sigma_{j=1}^{s_1} s_{1j,a}} = \prod_{j=1}^{s_1} U_{1j} P_{ID_{1j}}^{H_1\left(C_{1j,a}\| ID_{1j}\| T_{1j,a}\right)} \tag{4}$$

If the verification is successful, CN_{1a}decrypts the data $c_{1j} = Dec_{K_{a,1j}}(C_{1j})$, where $K_{a,1j} = S_{1a}(h_{1j})$. After collecting all ciphertexts from CN_{1j} within the specified time, CN_{1a}performs the local aggregation algorithm for region O_1 using Equation (5), where $\Sigma_{j=1}^{s_1} z_{1j} = 0$:

$$c_1 = \Sigma_{j=1}^{s_1} c_{1j} = \Sigma_{j=1}^{s_1} \left(m_{1j} + z_{1j} + v_{1j}\right) = \Sigma_{j=1}^{s_1} \left(m_{1j} + v_{1j}\right) \tag{5}$$

After completing local data aggregation, CN_{1a} initiates local consensus. Once local consensus is completed, the encrypted data from the region is sent to all ESs. Other regions $O_2, O_3, \ldots, O_w$ similarly perform local aggregation and consensus algorithms to achieve local data aggregation.

5.3. Results Global Aggregation

After CN_{1a} completes local aggregation, it encrypts the local aggregation result c_1 as $C_1 = EncK_{1a,i}(c_1)$ and sends it to ES_i. It then computes the signature and sends $Msg_{1a,i} = (C_{1a,i}||ID_{1a}||T_{1a,i}||U_{1a}||s_{1a,i})$ to ES_i.

After receiving $Msg_{1a,i}$, ES_i first checks the validity of the timestamp $T_{1a,i}$. After receiving $Msg_{ja,i}$ from all CN_{ja}, ES_i verifies the correctness of the signatures using Equation (6):

$$P^{\Sigma_{j=1}^{w} s_{ja,i}} = \prod_{j=1}^{w} U_{ja} P_{ID_{ja}}^{H_1\left(C_{ja,i}\| ID_{ja}\| T_{ja,i}\right)} \tag{6}$$

If the verification is successful, ES_i decrypts the data $c_1 = Dec_{K_{i,1a}}(C_1)$, where $K_{i,1a} = S_i(h_{1a})$. After collecting all ciphertexts from CN_{ia} within the specified time, ES_i performs the global aggregation algorithm using Equation (7):

$$c = \Sigma_{i=1}^{\mathrm{w}} c_i = \Sigma_{i=1}^{\mathrm{w}} \left(\Sigma_{j=1}^{s_i} (m_{ij} + v_{ij}) \right) \tag{7}$$

After completing global aggregation, ES_a initiates global consensus. Once global consensus is completed, the global aggregation result c is encrypted as $\mathrm{C} = EncK_{a,c}(\mathrm{c})$, where $K_{a,c} = S_a(h_c)$. ES_a then computes the signature and sends $Msg_{a,c} = (C_{a,c}||ID_a||T_{a,c}||U_a||s_{a,c})$ to the PC.

5.4. Data Decryption

After receiving $Msg_{a,c}$, the PC first checks the validity of the timestamp $T_{a,c}$. If valid, it verifies the signature using Equation (8):

$$P^{s_{a,c}} = U_a P_{ID_a}^{H_1(C_{a,c}\|ID_a\|T_{a,c})} \tag{8}$$

The PC then decrypts the aggregated ciphertext $c = DecK_{c,a}(C)$ and computes $P = \Sigma_{i=1}^{\mathrm{w}}(\Sigma_{j=1}^{s_i} S_c(h_{ij}))$ to recover the aggregated value $M = c - P = \Sigma_{i=1}^{\mathrm{w}} \Sigma_{j=1}^{s_i} m_{ij}$. Since $m_{ij} = \Sigma_{t=1}^{\mathrm{l}} m_{1j}^{(t)} \hat{A}_t = m_{1j}^{(1)} \hat{A}_1 + m_{1j}^{(2)} \hat{A}_2 + \cdots + m_{1j}^{(l)} \hat{A}_l$, we have: $M = \Sigma_{i=1}^{\mathrm{w}} \Sigma_{j=1}^{s_i} m_{ij} = \Sigma_{t=1}^{\mathrm{l}} \hat{A}_t (\Sigma_{i=1}^{\mathrm{w}} \Sigma_{j=1}^{s_i} m_{ij}^{(t)})$.

For $1 \le t \le l$, the PC can recover each component $M^{(t)} = Mmoda_t$ of the aggregated value M using the CRT. For example, $M^{(1)}$ is the aggregated value of the first component of the multidimensional data across all regions.

5.5. Dynamic CN Management

To enhance system robustness, the dynamic joining and exiting of CNs must be considered. Replacing a CN is equivalent to first exiting and then joining.

CN Joining: Assume a new CN with identity ID_{im} joins region O_i. First, CN_{im} applies for registration with the TA, which sends $\{x_{ID_{im}}, S_{im}(y)\}$ to CN_{im} through a secure channel and publishes $\{h_{im}, P_{ID_{im}}\}$. Other CNs in region O_i update their local encryption masks $z'_{ij} = z_{ij} + sgn(h_{ij} - h_{im}) S_{ij}(h_{im})$, $j = 1,2,\ldots,s_i$. Finally, CN_{im} computes its local encryption mask $z_{im} = \Sigma_{t=1}^{s_i} sgn(h_{im} - h_{it}) S_{im}(h_{it})$.

CN Exiting: Assume a CN with identity ID_{im} exits region O_i. Other CNs in region O_i update their local encryption masks $z'_{ij} = z_{ij} - sgn(h_{ij} - h_{im}) S_{ij}(h_{im})$, $j = 1,2,\ldots,s_i, j \neq m$.

5.6. Dual-Layer Consensus Algorithm

In this system, CNs are lightweight nodes divided into w regions based on geographical location. Each region performs local consensus on the local aggregation results using an improved Raft algorithm based on reputation values. All ESs are full nodes, and since their number is small and does

not require dynamic addition in the short term, they form a static structure. Therefore, the global consensus algorithm uses the PBFT algorithm.

The core of the Raft algorithm is to elect a leader node. In this system, the leader node CN_{ia} in each region is elected through reputation-based voting. The requirements are: first, to correctly complete local aggregation and send the aggregation results to the edge cloud server layer; second, to complete local consensus in the shortest possible time to avoid affecting global consensus. During initialization, the PC designates the leader node in each region. After the first local consensus, the reputation values of other nodes are generated as follows:

Taking region O_1 as an example, during the log replication phase, CN_{1a} broadcasts the sending time $time_{send}$ to other CN_{1j}. Upon receiving the acknowledgment message from CN_{1j}, CN_{1a} records the receiving time $time_{receive,j}$ and calculates the delay $time_j = time_{receive,j} - time_{send}$. The delay of each node is normalized, and the reputation values $value_j$ of other nodes are initialized in the range of 0 to 100, with the leader node's reputation value initialized to 110.

Reputation values are divided into four levels: A, B, C, and D, corresponding to 120-101, 100-71, 70-41, and 40-0, respectively. The leader node's reputation value is updated based on the order of completing local aggregation results. When all CN_{ia} in each region complete local aggregation, the timestamps are recorded and sorted in ascending order. The reputation values of the first x leader nodes are updated as $value_j = value_j^{pre} + 5$ and the last y leader nodes are updated as $value_j = value_j^{pre} - 5$, where $value_j^{pre}$ is the previous reputation value, and x, y are determined based on specific conditions. The following situations trigger a leader election in region O_i: ES_a does not receive local aggregation data from region O_i, region O_i fails to complete local data consensus, or a node's reputation value exceeds that of the leader node. During the new election, the node with the highest reputation value initiates the voting, and after the election, the reputation values of other nodes are initialized.

After completing consensus, ordinary nodes' reputation values are increased based on their reputation level using Equation (9):

$$\begin{cases} value_j^{pre} + 4 & value_j^{pre} > 100 \\ value_j^{pre} + 3 & 100 \geq value_j^{pre} > 70 \\ value_j^{pre} + 2 & 70 \geq value_j^{pre} > 40 \\ value_j^{pre} + 1 & value_j^{pre} \leq 40 \end{cases} \tag{9}$$

The penalty formula for the reputation value of ordinary nodes is as follows Equation (10):

$$\begin{cases} value_j^{pre} - 5 & Failure\ to\ upload\ data \\ value_j^{pre} - 3 & Consensus\ not\ participated\ in \end{cases} \tag{10}$$

6. PERFORMANCE EVALUATION

One of the design goals of this scheme is to ensure lightweight data aggregation, reducing the computational and storage overhead on both the charging pile and grid company sides. This section conducts comparative experiments on computational overhead, communication overhead, consensus efficiency, and security analysis to demonstrate the feasibility and practicality of the proposed scheme in resource-constrained environments.

The hardware used in the experiments is a laptop with an Intel(R) Core (TM) i7-8550U (1.99 GHz) processor, 8 GB of memory, and a 64-bit Windows 11 operating system. The JPBC library [22] is used to obtain the running time of cryptographic operations. For convenience of comparison, let $s_i = n$, indicating that each region has n CNs, with a total of w regions and nw CNs. The number of ESs is m, and the privacy data is l-dimensional.

6.1. Computational Overhead

The computational overhead of the scheme is mainly related to the encryption operations during the aggregation process. To simplify the representation, some symbols are introduced to represent related operations. The descriptions of the operations and their corresponding running times are shown in Table 2. The cost of hash operations is negligible compared to multiplication and exponentiation operations, so the computational cost of hash operations is not considered. To ensure a fair comparison, all schemes adopt the same security parameter, $\lambda = 80$. The running time for each operation is the average of 1000 executions.

Table 2. Runtime of Cryptographic Operations

Symbols	Descriptions	Runtime/ms
T_a	Time of addition operation on Z_q^*	0.0012
T_m	Time of multiplication operation on Z_q^*	0.0016
T_{eG}	Time of exponentiation operation on G	0.9754
T_{mG}	Time of modular multiplication operation on G	0.0096
T_{en^2}	Time of exponentiation operation on $Z_{N^2}^*$	3.985
T_{mn^2}	Time of multiplication operation on $Z_{N^2}^*$	0.0254
T_{bp}	The time of bilinear pairing operation	5.219

In the local aggregation process of this scheme, each CN needs to perform lT_m operations and $(l-1)T_a$ operations to pack multidimensional data into one-dimensional data. Then, it needs to perform $2T_a$ operations to encrypt and generate the ciphertext C, and T_{eG} operation to generate a digital signature. Additionally, the CN_{ia} in each region needs to perform $nT_{eG} + (2n-1)T_{mG}$ operations to batch verify the correctness of signatures within the region, while performing $(n-1)T_a$ operations for local data aggregation, and finally T_{eG} operation to generate a digital signature. In the global aggregation process, the ES performs $(w+1)T_{eG} + 2wT_{mG}$ operations to batch verify the correctness of signatures for local aggregation results, while performing $(w-1)T_a$ operations for global data aggregation. The ES_a needs an additional T_{eG} operation to generate the digital signature for the global aggregation data. The data requester performs $2T_{eG} + T_{mG}$ operations to verify the correctness of the global aggregation result and $T_a$operations to obtain the aggregated value M. The computational overhead of each entity is shown in Table 3.

In the EPPDA [7] scheme, each user needs $2T_{en^2} + T_{mn^2} + T_{eG} + 3T_m$operations to encrypt and sign local data. All blockchain devices require $2T_{en^2} + 2T_{mn^2} + 2nwT_{eG} + 3nwT_m + 2nwT_a$ operations to confirm the correctness of signatures and aggregate data. The data requester needs T_{en^2} operations to decrypt and obtain the aggregated data. For convenience of comparison, blockchain devices correspond to the ES in this scheme, and the total computational overhead of blockchain devices is averaged across each ES.

Table 3. Computational Overhead of Different Entities

Entity	Computational Overhead
CN_{ij}	$lT_m + (l+1)T_a + T_{eG}$
CN_{ia}	$lT_m + (l+n)T_a + (n+1)T_{eG} + (2n-1)T_{mG}$
ES_i	$(w+2)T_{eG} + 2wT_{mG} + (w-1)T_a$
ES_a	$(w+3)T_{eG} + 2wT_{mG} + (w-1)T_a$
PC	$2T_{eG} + T_{mG} + T_a$

In the PATM [8] scheme, each user needs $3T_{eG} + T_{mG}$ operations to encrypt and sign local data. The gateway node requires $(n+2)T_{eG} + 3nT_{mG}$ operations to confirm the correctness of signatures and aggregate data to complete the signature. The edge cloud server requires $2wT_{eG} + (w+d+1)T_{mG}$ operations to confirm data integrity and obtain the aggregated data, where the PATM [8] scheme uses a secret sharing algorithm, and here we assume $d = 2m/3$. For convenience of comparison, the gateway node corresponds to the CN_{ia} in this scheme. Table 4 compares the computational overhead of the proposed scheme with the other two schemes.

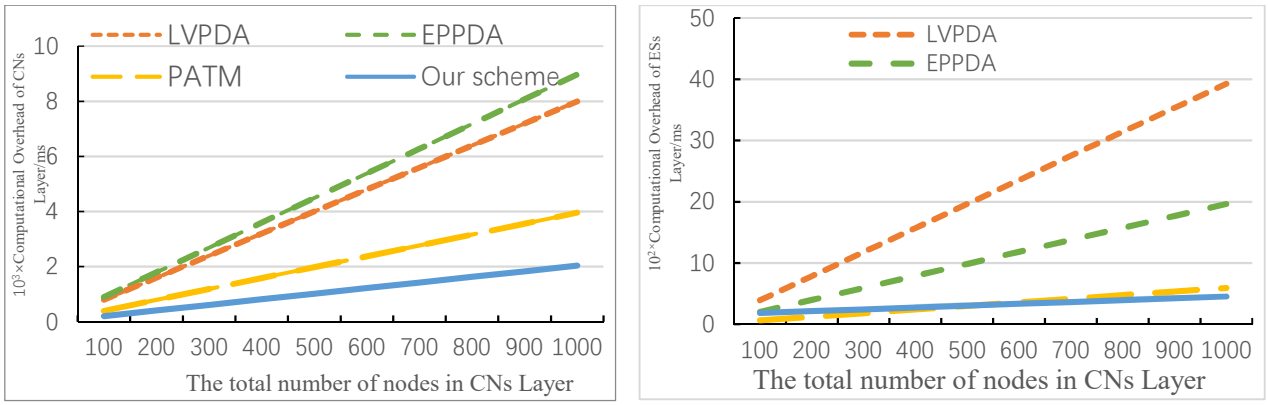
In the LVPDA [23] scheme, each user needs $2T_{en^2} + 3T_{mG}$ operations to encrypt and sign local data. The edge cloud server requires $4nwT_{eG} + nwT_{mn^2}$ operations to confirm data integrity and obtain the aggregated data. The data requester needs $2T_{mn^2} + 2T_{bp}$ operations to decrypt and obtain the aggregated data. Table 4 compares the computational overhead of the proposed scheme with the other two schemes.

Table 4. Comparison of Computational Expenses

Participant	EPPDA [7]	PATM [8]	LVPDA [23]	Our scheme
CN_{ij}/ms	8.9756	2.9358	7.9988	0.0028l+0.9778
CN_{ia}/ms	-	1.0042n+1.9508	-	0.0028l+0.9958n+0.9574
ES_i/ms	(8.0208+1.958nw)/m	1.9604w+0.0064m+0.0096	3.927nw/m	0.9958w+1.9688
ES_a/ms	-	-	-	0.9958w+2.9442
PC/ms	3.985	-	10.4888	1.9616

For ease of comparison, let the dimension of charging data $l = 20$, the number of CNs in each region $n = 100$, and the number of ESs $m = 30$. The total computational overhead of the charging pile layer is shown in Figure 4(a). With a total of 1000 nodes in the charging pile layer, the proposed scheme reduces the computational overhead by 77.2%, 48.5%, and 74.5% compared to EPPDA [7], PATM [8], and LVPDA [23], respectively. The total computational overhead of the edge cloud server layer is shown in Figure 4(b).

The proposed scheme reduces the computational overhead by 81.7%, 39.7%, and 90% compared to EPPDA [7], PATM [8], and LVPDA [23], respectively. On the data requester side, in the PATM [8] scheme, the PC, as a node in the blockchain, can directly query the blockchain to obtain the aggregated data. The proposed scheme is significantly lower than EPPDA [7] and LVPDA [23], and since it does not require saving the blockchain ledger, it reduces storage overhead. In summary, the proposed scheme reduces computational overhead in both the charging pile layer and the edge cloud server layer, and the computational overhead on the data requester side is within an acceptable range, without consuming additional storage overhead.



(a) **(b)**

Figure 4. (a) Computational Overhead of CNs Layer; (b) Computational Overhead of ESs Layer

6.2. Communication Overhead

The communication overhead of the scheme is mainly determined by the message length and the number of communications. To simplify the representation, some symbols are introduced to represent the length of encrypted elements. The descriptions of the elements and their corresponding lengths are shown in Table 5.

Table 5. Length of the Element

Symbol	Descriptions	Length/bits
$\|G\|$	Length of elements in G	160
$\|Z_q^*\|$	Length of elements in Z_q^*	160
$\|Z_{N^2}^*\|$	Length of elements in $Z_{N^2}^*$	1024
$\|T\|$	Length of timestamp	32
$\|ID\|$	Length of ID	32

In the local aggregation phase of this scheme, CN_{ij} sends $Msg_{ij,a} = \left(C_{ij,a} \left|\left| ID_{ij} \right|\right| T_{ij,a} \left|\left| U_{ij} \right|\right| s_{ij,a}\right)$ to the master node CN_{ia} in the region, with a communication overhead of $2|Z_q^*| + |G| + |T| + |ID|$ bits. Since the data structure of communication between nodes is the same, the communication overhead when CN_{ia} sends to ES_i is also $2|Z_q^*| + |G| + |T| + |ID|$ bits. The communication overhead of the other three schemes is compared in Table 6.

Table 6. Communication Overhead

Entity	CN to CN	CN to ES
EPPDA[7]	-	$\|Z_{N^2}^*\| + \|G\| + \|T\| + \|ID\|$
PATM[8]	$\|Z_q^*\| + 3\|G\| + \|T\| + \|ID\|$	$\|Z_q^*\| + 3\|G\| + \|T\| + \|ID\|$
LVPDA[23]	-	$2\|Z_q^*\| + 5\|G\| + \|T\| + \|ID\|$
Our scheme	$2\|Z_q^*\| + \|G\| + \|T\| + \|ID\|$	$2\|Z_q^*\| + \|G\| + \|T\| + \|ID\|$

This section compares the total communication overhead of completing local and global aggregation in different schemes. In the proposed scheme, completing local aggregation requires $(n-1)w$ communications, while the PATM [8] scheme requires nw communications. Completing global aggregation in the proposed scheme requires mw communications, while the PATM [8] scheme requires mw communications, the EPPDA [7] scheme requires $nw+m-1$ communications, and the LVPDA [23] scheme requires nw communications. Let the number of CNs in each region $n=100$, and the number of ES nodes $m=30$. The total communication overhead is compared in Figure 5. With a total of 1000 nodes in the charging pile layer, the proposed scheme reduces the communication overhead by 30.4%, 40.8%, and 45.3% compared to PATM [8], LVPDA [23], and EPPDA [7], respectively.

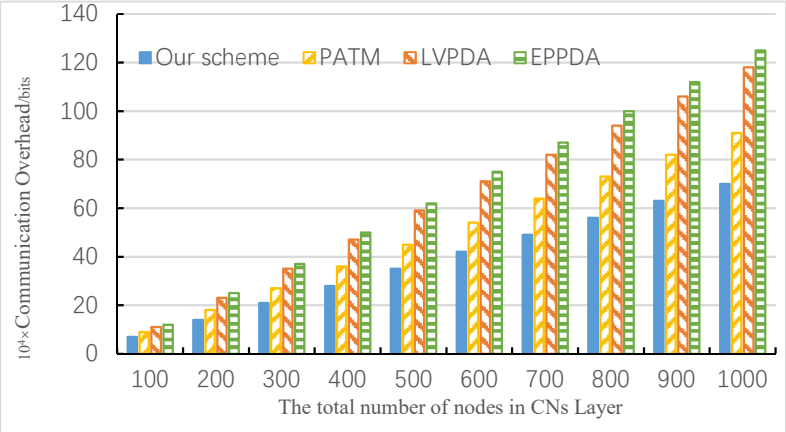


Figure 5. Communication Overhead

6.3. Consensus Efficiency

This section uses the Go language to simulate the communication process of consensus nodes by opening multiple threads. The consensus efficiency of different consensus algorithms is compared by analysing the consensus delay of the algorithms. The PATM [8] scheme uses the Proof of Work (PoW) consensus algorithm, while the EPPDA [7] scheme uses the Controlled Proof of Work (cPoW) consensus algorithm, which takes an average of 0.3 to 0.4 minutes to solve lightweight mining problems [7], and its consensus efficiency is better than the PoW consensus algorithm. Additionally, the commonly used Raft consensus algorithm [6] and the PBFT consensus algorithm [15] are compared. Consensus delay refers to the total time from when the client submits a transaction request to when the confirmation is completed, calculated using Equation (11):

$$DT = time_{req} - time_{reply} \tag{11}$$

In the proposed scheme, the charging pile layer performs the improved Raft consensus algorithm based on reputation values in each region, while the edge cloud server layer uses the PBFT consensus algorithm. The average consensus delay is obtained from multiple experiments. The comparison results are shown in Figure 6. The Raft algorithm has the lowest consensus delay among the compared algorithms but cannot resist Byzantine attacks. The PBFT algorithm has the highest consensus delay among the compared algorithms, and the delay increases rapidly as the number of nodes increases. The proposed scheme combines the two methods, significantly reducing the consensus delay compared to the PBFT algorithm and the cPoW consensus algorithm of the EPPDA [7] scheme, and the delay does not increase rapidly with the number of nodes. This is suitable for the characteristics

of a large number of CNs that are prone to changes, ensuring low latency in the charging pile layer while resisting Byzantine attacks in the edge cloud server layer.

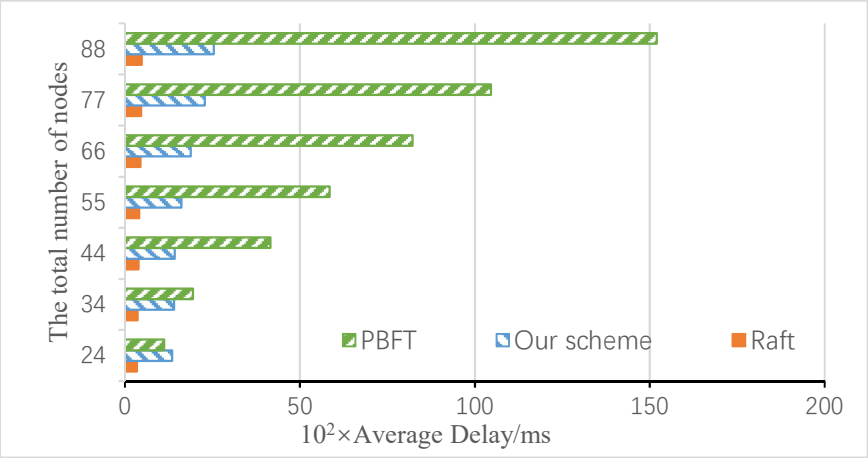


Figure 6. Consensus Latency

6.4. Security Analysis

This section analyzes the security comparison between the proposed scheme and the EPPDA [7], PATM [8], and LVPDA [23] schemes, as shown in Table 7. In the proposed scheme, the data transmitted on public channels are symmetrically encrypted, and only the communicating parties can decrypt them. Additionally, mask encryption is used, ensuring that attackers cannot infer the private information of specific CN_{ij}. The charging pile layer uses the Raft consensus algorithm. When no more than half of the CNs in a region fail to upload their charging data on time due to network issues or their own faults, the master node can still complete local consensus and upload the local aggregation data. The proposed scheme adopts a dual-layer consensus algorithm, ensuring decentralization while the edge cloud server layer can resist Byzantine attacks. The charging pile layer can verify the correctness of local aggregation data. The proposed scheme uses an aggregate signature algorithm, ensuring lightweight integrity verification.

Table 7. Comparison of Security

Participant	EPPDA[7]	PATM [8]	LVPDA [23]	Our scheme
Privacy Protection	√	√	√	√
Robustness	×	√	√	√
Decentralization	√	√	×	√
Byzantine Attack-Resistant	×	×	×	√
Lightweight Verification	×	√	×	√
Charging Pile End Inspection	×	×	×	√

7. CONCLUSION

This paper proposes a multidimensional privacy data aggregation scheme for charging piles. Firstly, non-interactive symmetric encryption is more suitable for charging piles with limited computational resources compared to homomorphic encryption. The Chinese Remainder Theorem is used to

package multidimensional data, reducing communication overhead, while the aggregate signature scheme improves batch verification efficiency. Secondly, consortium blockchain is used to address data centralization, avoiding single-point failure issues and improving data security. A dual-layer consensus algorithm is proposed based on the system model, prioritizing consensus efficiency and system robustness at the charging pile layer, while the edge cloud server layer can resist Byzantine attacks. Finally, experimental comparisons with other schemes demonstrate that the proposed scheme reduces computational and communication overhead by at least 48.5% and 30.4%, respectively, proving the lightweight nature of the proposed scheme.

CONFLICTS OF INTEREST

The authors declare that they have no conflict of interest.

ACKNOWLEDGEMENTS

This research was funded by the National Natural Science Foundation of China [grant number 62171185].

REFERENCES

[1] Zhao Xiaolei; LI Xuemei. Empirical evidence of the impact of industrial policies on the charging infrastructure industry and their mechanisms[J]. China Population, Resources and Environment, 2024,34(07):47-57.

[2] Zheng Xueqin. As of October 2024, the cumulative number of charging facilities reached 11.88 million units[J]. Auto Review,2024,(12):116-117

[3] Jiang Linru,Long Yi,Li Xingyuan, et al. Charging load analysis of multi-type electric vehicle based on measured data[J]. Electrical Measurement & Instrumentation,2023,60(1):36-4147

[4] Tang Xianghua;Zhu Fuyun;Lv Shuaishuai, et al. Design and research of new energy vehicles intelligent charging pile[J]. China's High-Tech, 2020(10):26-28

[5] X. Zuo, L. Li, H. Peng, et al. Privacy-Preserving Multidimensional Data Aggregation Scheme Without Trusted Authority in Smart Grid. IEEE Systems Journal, 2021,15(1):395-406.

[6] Chen Xiuqiang,Wang Feng,Mao Guojun, et al. Privacy-preserving data aggregation scheme for smart grid based on blockchain[J]. Computer Engineering and Design, 2024,45(05):1343-1350

[7] Azhar Mahmood, Abid Khan, Adeel Anjum, et al. An efficient and privacy-preserving blockchain-based secure data aggregation in smart grids. Sustainable Energy Technologies and Assessments,2023,60:103414.

[8] C. Hu, Z. Liu, R. Li, et al. Smart Contract Assisted Privacy-Preserving Data Aggregation and Management Scheme for Smart Grid, IEEE Transactions on Dependable and Secure Computing, 2024,21(4): 2145-2161.

[9] Li Wenjin,Cai Ying,Fan Yanfang, et al. Privacy-preserving Data Aggregation Scheme in Vehicular CrowdSensing[J/OL], Journal of Chinese Computer Systems. 2025, 46(1): 200-208

[10] K. Xue, B. Zhu, Q. Yang, et al. An Efficient and Robust Data Aggregation Scheme Without a Trusted Authority for Smart Grid, IEEE Internet of Things Journal,2020(7)3:1949-1959.

[11] Y. Su, J. Li, Y. Li and Z. Su. Edge-Enabled: A Scalable and Decentralized Data Aggregation Scheme for IoT, IEEE Transactions on Industrial Informatics,2023(19)2:1854-1862.

[12] Zhang Yong;Zhang Shengfa;Liu Dengzhi. Secure Multidimensional Data Aggregation with Secret Sharing in Wireless Sensor Networks[J], Chinese Journal of Sensors and Actuators. 2024,37(04):709-715

[13] M. A. Mustafa, S. Cleemput, A. Aly, et al. A Secure and Privacy-Preserving Protocol for Smart Metering Operational Data Collection, IEEE Transactions on Smart Grid, 2019(10)6:6481-6490.

[14] Y. -C. Chen, J. -K. Yang, H. -C. Yen , et al. Dual-Cloud Multi-Secret Sharing Architecture for Privacy Preserving Persistent Computation, IEEE Transactions on Information Forensics and Security, 2024(19):7523-7535.

[15] Li Da;Feng Jingli;Ping Jian , et al. rustworthy Aggregation Method of Electric Vehicle Charging Private Data Based on Double-Layer Blockchain, Electric Power Construction, 2023,44(11):13-22

[16] Loukil, F.; Ghedira-Guegan, C.; Boukadi, K, et al. Privacy-Preserving IoT Data Aggregation Based on Blockchain and Homomorphic Encryption, Sensors 2021(21):2452.

[17] Shafeeq, S. and Fischer, M. SEBDA: A Secure and Efficient Blockchain Based Data Aggregation Scheme. International Conference on Security and Cryptography - SECRYPT 2023:369-376.

[18] M. Zhang, Y. Li, Y. Ding, et al. A Lightweight and Robust Multidimensional Data Aggregation Scheme for IoT, IEEE Internet of Things,2024(11)2:3578-3588.

[19] Tan Pengliu;Wang Runshu;Zeng Wenhao, et al. Overview of Blockchain Consensus Algorithms[J], Computer Science, 2023,50(S1):691-702

[20] Ongaro, Diego and John K. Ousterhout. In Search of an Understandable Consensus Algorithm. USENIX Annual Technical Conference Philadelphia, PA,2014: 978-1-931971-10-2.

[21] Yuan Haotian, Li Fei. Double layer consensus algorithm based on improved Raft consensus algorithm and PBFT [J]. Application Research of Computers, 2024, 41 (5): 1314-1320

[22] Lynn B. PBC library [EB/OL]. [2022-08-15]. http://crypto.stanford.edu/pbc/.

[23] J. Zhang, Y. Zhao, J. Wu, et al. LVPDA: A Lightweight and Verifiable Privacy-Preserving Data Aggregation Scheme for Edge-Enabled IoT, IEEE Internet of Things,2020(7)5:4016-4027