An Efficient and Robust Multidimensional Data Aggregation Scheme for Smart Grid Based on Blockchain

Lin You, *Member, IEEE*, Xinhua Zhang, Gengran Hu, Longbo Han

***Abstract*— In order to analyze real-time power data without revealing user's privacy, privacy-preserving data aggregation has been extensively researched in smart grid. However, most of the existing schemes either have too much computation overhead and cannot achieve dynamic users, or require a trusted center. In this paper, we propose an efficient and robust multidimensional data aggregation scheme based on blockchain. In our scheme, a leader election algorithm in Raft protocol is used to select a mining node from all smart meters to aggregate data. A dynamically verifiable secret sharing homomorphism scheme is adopted to realize flexible dynamic user management. In addition, our scheme can not only resist internal and external attackers but also support multidimensional data aggregation and fault tolerance. Compared with other schemes, our scheme not only supports user fault tolerance, but also supports fault tolerance of the intermediate aggregation node. The security analysis shows that our proposed scheme is IND-CPA secure and can meet stronger security features. Our performance analyses show that compared with other schemes, our scheme can be implemented with lower computation cost and communication overhead.**

***Index Terms*— Privacy-preserving, Smart Grid, Blockchain, Secret Sharing Homomorphism**

I. Introduction

With the development of the Internet of Things, smart meters have been widely used. Smart meters can not only charge bills, but also report real-time consumption data and other information to the utility provider, and then the utility provider can perform dynamic pricing and data analysis based on these information. But real-time data may reveal the users' personal behavior. Therefore, in order to protect users' privacy, data aggregation schemes are usually adopted for smart grid, so that the utility provider can only obtain total electricity consumption data, but cannot obtain real-time data of individual user [1].

Although the data aggregation scheme can prevent the utility provider from obtaining real-time data of a single user and protect users' privacy, there are still some other problems that need to be resolved in practical application [2]. First of all, in some existing schemes [3] [4] [5] [6] [7] [8] [9] [10], a trusted third-party is required to generate a series of secret parameters for the system to encrypt the real-time data to ensure users' privacy. However, in practical application, we generally don't want to introduce or find it difficult to find a trusted third party.

Secondly, in some schemes [3] [4] [5] [11] [6] [8] [12] [13] [14] [15] [16], smart meters can only report the total data consumed by all electrical appliances, but the utility provider may require data of multiple types of electrical appliances for in-depth analysis.

Thirdly, some schemes [9] [10] [14] [15] [16] [17] [18] [19] [20] use homomorphic encryption to encrypt real-time data, but this will bring huge computing overhead, which is a great challenge for smart meters with limited computing power.

Finally, we also need to consider the scalability of the system. In some schemes, after the system is deployed, it is impossible to add or delete a smart meter for the system or it is expensive to do so, or if some of the deployed smart meters are damaged, the entire system cannot work.

A. Our Contributions

In this paper, a multidimensional data aggregation scheme based on Blockchain (BBMDA) is proposed, which can solve the above problems well. Our contributions can be summarized as follows.

1. We propose an efficient data aggregation scheme based on blockchain. The secure and lightweight operation based on the mask scheme ensures the efficiency of our scheme. Our scheme does not require a trusted third-party, and can realize the fault tolerance of intermediate node.

2. Our scheme also introduces a dynamically verifiable (t, n) secret sharing scheme to achieve flexible dynamic user management and user fault tolerance without trusted authority.

3. Our scheme enables the smart meter to report multidimensional data to the utility provider. Therefore, the utility provider can conduct in-depth analysis of these data of multiple types.

4. We adopt a signature scheme with batch verification to allow intermediate node to effectively verify data integrity.

B. Organization of the Paper

The remainder of this paper is organized as follows. In Section II and III, the preliminaries and system model are introduced, respectively. Our scheme is presented in Section IV,

This work was supported in part by the National Natural Science Foundation of China under Grant 61772166 and in part by the Key Program of the Nature Science Foundation of Zhejiang province of China under Grant LZ17F020002.(*Corresponding Author: Lin You*)

Lin You, Xinhua Zhang, Gengran Hu and Longbo Han are with the School of Cyberspace, Hangzhou Dianzi University, 310018, China (e-mail: mryoulin@gmail.com, zxhua, grhu@hdu.edu.cn, longbohan@hdu.edu.cn).

followed by its security analysis and performance evaluation in Section V and VI. Finally, in Section VII, we concludes this paper.

II. Related Work

In recent years, many effective data aggregation schemes have been proposed to protect users' privacy for smart grid. The homomorphic encryption system is one of the most used methods such as Paillier homomorphic encryption. Chen et al. [19] used Paillier homomorphic encryption to report data of multiple types in a reporting message, so the utility provider can perform the variance analysis and the one-way analysis of variance on the data. In [14], the (t, n) threshold secret sharing scheme is used to realize flexible dynamic user management, but the scheme only considers aggregating a single type of data. Ming et al. [20] proposed an efficient scheme based on elliptic curves ,which used superincreasing sequence to aggregate multidimensional data. However the scheme requires a trusted center to distribute security keys. In [9], dynamic users can be efficiently implemented, but this scheme does not consider user authentication and data integrity checking. In [10], A fault-tolerant scheme is proposed, which uses Paillier homomorphic encryption algorithm to encrypt data and supports batch authentication of intermediate aggregators, but this scheme also requires a trusted center. Liu et al. [16] proposed a scheme that does not require a trusted center. However, this scheme cannot aggregate multidimensional data and does not support dynamic users.

Homomorphic encryption can protect data privacy well, but it brings a huge computational burden to smart meters. Therefore, in order to improve efficiency, some masking-based schemes have been proposed. F. Knirsch et al. [8] proposed a mask-based spatio-temporal aggregation scheme, which can support fault tolerance. In A. Alsharif et al. [12], the data is masked by the masking value shared by each smart meter and the agent randomly selected from them. After all users' data are aggregated, the sum of the mask will be 0. Karampour et al. [**?**] proposed a scheme that can resist collusion attack of n-2 users, In the scheme, each user uses AV-net mask to encrypt data, and the sum of AV-net mask is 0, but their scheme only considers aggregating a single type of data.

Since most of the existing schemes are centralized architectures with single point of failure and inaccurate feedback problem, considering the characteristics of decentralization and distributed storage of blockchain, the application of blockchain in the smart grid can solve the above problems well [13] [11] [**?**]. Guan et al. [13] proposed an scheme which is based on blockchain. They divide users into different groups, each group has a bloom filter to verify users' identity and a private blockchain to record data, but users' data is transmitted in plaintext. Fan et al. [11] proposed a decentralized privacy-preserving data aggregation (DPPDA) scheme, and the Paillier homomorphic cryptosystem and the Boneh-Lynn-Shacham short signature are adopted to ensure the confidentiality and integrity of user data. However, this scheme only focused on single-dimensional data aggregation.

III. Preliminaries

A. Blockchain

Blockchain, a distributed append-only public ledger technology, was first proposed in 2008 by Satoshi Nakamoto for Bitcoin [21]. A complete blockchain system contains many technologies, including blocks storing data and digital signatures above them, timestamp, Merkle tree, P2P network and other technologies, as well as consensus algorithms for maintaining the system. It can solve the problem of single point of failure of the current centralized structure by applying distributed characteristics of blockchain to IoT network. Therefore, many studies have applied blockchain to IoT [22].

Raft [23] is a consensus algorithm for managing a replicated log. In Raft protocol, first a distinguished leader is elected, then the leader is given full rights to manage accounting. The leader receives the accounting request from the client, completes the accounting operation, generates a block, and replicates it to other accounting nodes. When the leader does not work, a new leader is elected.

B. (t, n) Threshold Verifiable Secret Sharing Homomorphism Scheme

1) Pedersen's Verifiable Secret Sharing Scheme: In order to verify the correctness of secret share, the verifiable secret sharing scheme is proposed. The first information-theoretic security non-interactive verifiable secret sharing scheme was proposed by Pedersen [24].

Let g and h be elements of G_q, G_q is the unique subgroup of $\mathbb{Z}_p^*$ of order q , such that nobody knows $\log_g h$. In fact, $\mathbb{Z}_q$ is a field, the dealer D can distribute $s \in \mathbb{Z}_q$ as follows:

1. D randomly chooses $u \in \mathbb{Z}_q$, and publishes a commitment to $s : E_0 = E(s, u) = g^s h^u$.

2. D chooses $F \in \mathbb{Z}_q[x]$ of degree at most $t-1$ satisfying $F(0) = s$, and computes $s_i = F(i)$ for $i = 1, \ldots, n$. Let $F(x) = s + F_1 x + \cdots + F_{t-1} x^{t-1}$, D randomly chooses $G_1, \ldots, G_{t-1} \in Z_q$ and uses G_i when committing to F_i for $i = 1, \ldots, t-1$. D broadcasts

$$E_i = E(F_i, G_i) \tag{1}$$

for $i = 1, \ldots, t-1$.

3. Let $G(x) = u + G_1 x + \cdots + G_{t-1} x^{t-1}$ and let $u_i = G(i)$ for $i = 1, \ldots, n$. Then D sends (s_i, u_i) secretly to P_i for $i = 1, \ldots, n$. When P_i receives his share (s_i, u_i), he verifies that

$$E(s_i, u_i) = \prod_{j=0}^{t-1} E_j^{i^j} \tag{2}$$

4. Utilize Lagrange interpolation formula as (3), the secret s can be reconstructed by any party who collects t or more different shares, and $s = F(0)$.

$$F(x) = \sum_{i=1}^{t} F(i) \prod_{j=1, j \neq i}^{t} \frac{x-j}{i-j} \tag{3}$$

2) Secret sharing homomorphism: Secret sharing homomorphism was introduced by Benaloh [25]. Assume there are two secret s_1, s_2 and they are shared by two polynomials $f(x)$ and $g(x)$.

1. The dealer D sends the share $f(i)$ and $g(i)$ to the corresponding user P_i for $i = 1, \ldots, n$.

2. P_i computes and sends $f(i) + g(i)$ to D for $i = 1, \ldots, n$, $f(i) + g(i)$ can be regarded as the share corresponding to the secret $s_1 + s_2$.

3. Any party who collects t or more different shares can reconstruct secret $s = s_1 + s_2$ due to the additive homomorphism.

3) Extension to Support Dynamic Secret: In this article, the secret will change when the user joins or leaves. Therefore, according to [**?**], we design a method to support dynamic secret sharing.

Firstly, the dealer D randomly chooses a static secret ss and dynamic secret S, and shares ss to n participants by (t, n) secret sharing scheme. Then, D computes Ts as(5), and publishes Ts

$$Ts = S - ss \bmod q \tag{4}$$

To obtain the secret S, any party first reconstructs the static secret ss, then computes

$$S = Ts + ss \bmod q \tag{5}$$

When the dealer needs to update the secret, it only needs to compute Ts' as (5), then the new secret S' can be obtained by computing (6).

C. Chinese Remainder Theorem

In this section, we firstly briefly describe the Chinese Remainder Theorem [**?**]. Suppose that $q_1, q_2, ..., q_k$ are pairwise relatively prime positive integers, and let $a_1, a_2, ..., a_k$ be integers. Then, the system of congruences, $x \equiv a_j \bmod q_j$, for $1 \leq j \leq k$, has a unique solution modulo $Q = q_1 q_2 \cdots q_k$, which is given by

$$x = a_1 Q_1 Q_1^{-1} + \cdots + a_k Q_k Q_k^{-1} \bmod Q \tag{6}$$

where

$$Q_j = \frac{Q}{q_j}\ ,\ Q_j Q_j^{-1} \equiv 1 (\bmod\ q_j)\ ,\ 1 \leq j \leq k.$$

IV. System Model

A. Network Model

In our scheme, there are three entities including the smart meter (SM), the mining node (Mn) and the control center (CC), as Fig.1.

1. SM: It is the intelligent device of each user, which is used to collect data generated by the user. When a smart meter is produced, a unique identity ID will be registered in the blockchain.

2. Mn: In the system initialization stage, all smart meters use a leader election algorithm to select a smart meter as the mining node. Mn replaces the aggregator in the traditional model to verify the legitimacy of the transmitted data and aggregate encrypted data.

3. CC: It reads the aggregated data from Mn through the blockchain, and performs data analysis. In addition, CC also performs system initialization.

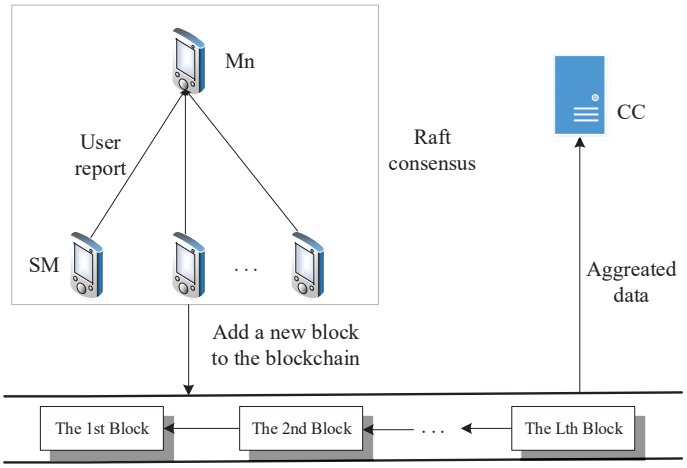


Fig. 1: System Model.

B. Threat Model

The attacker may be an external adversary or an internal network node, such as SM and CC. External attacker A may eavesdrop on the user's data and try to get plaintext data from ciphertext. A can also perform some active attacks, for example, hacking into CC's database to steal power data or tamper with encrypted data, or even replay valid packets that have been used. For internal attackers, they also try to derive individual electricity consumption from the aggregated data. Generally, users will follow the defined protocol and will not tamper with electricity data. In addition, we assume that there are at most $t-1$ malicious users colluding with each other in the system.

C. Design Goals

In order to defend against defined threats, our scheme should meet the following security requirements:

1. Confidentiality: The attacker cannot extract the individual real-time electricity data from the intercepted ciphertext.

2. Integrity: If the transmitted data is modified, it can be detected by authorized receivers.

3. Privacy Preservation: No one can obtain personal real-time data except oneself.

4. Identity Authentication: If an unregistered opponent joins and sends false data to the system, it can be discovered.

5. Dynamic User Management: The proposed scheme supports a user to dynamically join/exit the smart grid system without redoing complex initialization work.

6. Fault Tolerance: Since some smart meters may be malfunctioning and intermediate node could be compromised by the adversary, CC should still be able to obtain the aggregated data.

7. Decentralization: A trusted third party or central authority is not required in our scheme.

8. Forward Secrecy: It is required that the leakage of the current security key will not affect the confidentiality of previous personal information.

9. Resistance against Attacks: The smart grid can be subject to internal attacks and various external attacks such as modification attack, replay attack, impersonation attack and man-in-the-middle attack.

D. Security Assumption

(Elliptic Curve Computational Diffie-Hellman (ECCDH) Assumption). Consider a q order group $\mathbb{G}$, P is a generator of $\mathbb{G}$, for any $a, b \in [0, q-1]$, given aP, bP, there is an adversary A that computes abP with the advantage

$$Adv_A^{ECCDH} = Pr[abP \leftarrow A(\mathbb{G}, P, aP, bP)] \tag{7}$$

We say that the ECCDH assumption holds if the advantage Adv_A^{ECCDH} is negligible for any probabilistic polynomial time (PPT) adversary A under the security prameter 1^λ.

V. Our Proposed Scheme

In the section, we introduce our BBMDA scheme, some notions are gived in Table I.

TABLE I Notations

Notations	Descriptions
ID_i	Identity of user i
SM_i	The smart meter of user i
m_{ij}	The j-th dimension data of user i
p, q	Two large prime
$\mathbb{G}$	q order group
g, h	Elements of $\mathbb{G}$
P	A generator of $\mathbb{G}$
H_1, H_2	Secure hash functions $H_1, H_2 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$
x_i, x, PK_i, P_{pub}	Private key and public key of user i and Mn
q_j, Q, α_j, k_1	Public parameters of Chinese Remainder Theorem
SS_i	Static secret of user i
S_i	Dynamic secret of user i used to encrypt data
Ts_i	The parameter used to recover dynamic secret
s_{ij}, t_{ij}	Verifiable share issued by user i to user j
(t, n)	n shares, at least t shares can recover the secret
φ_{ij}, ψ_{ij}	Parameters of Pedersen's Verifiable Secret Sharing
c_i	Encrypted multidimensional data of user i
(R_i, z_i)	Signature generated by user i
M_j	Sum of the j-th dimension data of all users
X	The maximum value of electricity data

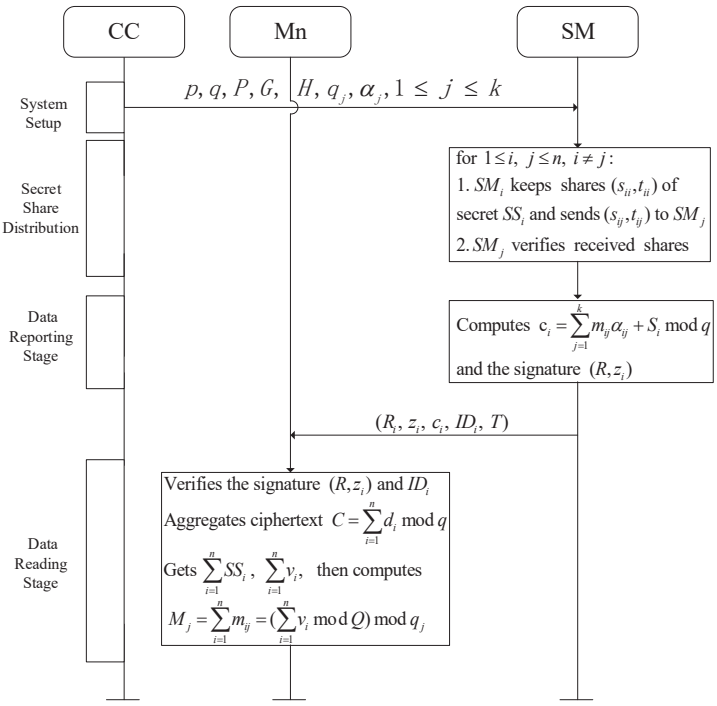


Fig. 2: Proposed Multidimensional Data Aggregation (BBMDA) Scheme.

A. Outline

In this section, we first give the overview of our BBMDA scheme. It consists of four stages: system initialization stage, secret share distribution stage, data reporting stage, and data reading stage (as Fig.2). Finally, the dynamicuser management and fault tolerance are proposed.

B. System Initialization

Supposing there are n users and ID of smart meter of each user is registered in the blockchain. Each smart meter uses the leader election algorithm in Raft protocol to select a smart meter as the mining node (Mn).

CC generates a q order group $\mathbb{G}$, which is based on the elliptic curve E defined on the finite field F_p, and P is the generator. Then, CC randomly selects k prime numbers $q_1, q_2, \ldots, q_k$, $|q_j| = k_1, k_1 \cdot (k+1) + \log_2 k < |q|$, and computes

$$\begin{cases} Q = q_1 q_2 \cdots q_k \\ Q_j = \frac{Q}{q_j},\ Q_j Q_j^{-1} \equiv 1 \bmod q_j \\ \alpha_j = Q_j Q_j^{-1} \end{cases} \tag{8}$$

where $Q_j^{-1} \in \mathbb{Z}_{q_j}^*$.

Finally, CC randomly chooses $g, h \in \mathbb{G}_q$, secure hash functions: $H_1 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$ and $H_2 : \{0,1\}^* \longrightarrow \mathbb{Z}_q^*$, then publishes the system parameters $\{p, q, P, \mathbb{G}, H_1, H_2, q_j, \alpha_j : j = 1, \ldots, k\}$.

In addition, each SM_i randomly selects $x_i \in \mathbb{Z}_q^*$ as his private key, and computes the public key $PK_i = x_i \cdot P$, then publishes (PK_i, ID_i). Mn selects a random number $x \in \mathbb{Z}_q^*$ as his key, and computes $P_{pub} = x \cdot P$.

C. Secret Share Distribution

1. Parameter generation

SM_i randomly selects $T_i, SS_i \in \mathbb{Z}_q^*$, and publishes the commitment to SS_i : $E_{i0} = E(SS_i, T_i) = g^{SS_i} h^{T_i}$. Then SM_i randomly selects $\varphi \in \mathbb{Z}_q^*[x]$ of degree at most $t-1$ satisfying $\varphi_i(0) = SS_i$. Let $\varphi_i(x) = \varphi_{i0} + \varphi_{i1}x + \cdots + \varphi_{i(t-1)}x^{t-1}$, SM_i chooses $\psi_{i1}, \ldots, \psi_{i(t-1)} \in \mathbb{Z}_q^*$ at random and uses ψ_{is} when committing to φ_{is} for $i = 1, \ldots, n, s = 0, \ldots, t-1$. Let $\psi_i(x) = \psi_{i0} + \psi_{i1}x + \cdots + \psi_{i(t-1)}x^{t-1}$, where $t \leq n$. Finally, SM_i broadcasts the commitment

$$E_{is} = E(\varphi_{is}, \psi_{is}) = g^{\varphi_{is}} h^{\psi_{is}} \tag{9}$$

for $i = 1, \ldots, n, s = 0, \ldots, t-1$.

2. Share distribution

SM_i computes $s_{ij} = \varphi_i(j), t_{ij} = \psi_i(j)$, for $j = 1, \ldots, n$. Then SM_i keeps (s_{ii}, t_{ii}) and securely sends (s_{ij}, t_{ij}) to SM_j as its verifiable secret share (as Fig.3), where $i, j = 1, \ldots, n, j \neq i$.

3. Share verification

After receiving his verifiable shares (s_{ij}, t_{ij}), SM_j verifies

$$E(s_{ij}, t_{ij}) = \prod_{s=0}^{t-1} E_{is}^{j^s} \tag{10}$$

for $i = 1, \ldots, n, i \neq j$.

If all shares are correct, for $j = 1, \ldots, n, SM_j$ computes $s_j = \sum_{i=1}^{n} s_{ij}$, $t_j = \sum_{i=1}^{n} t_{ij}$ and the commitment $(E_0, E_1, \ldots, E_{t-1})$ corresponding to secret SS $(SS = SS_1 + SS_2 + \cdots + SS_n)$ as

$$E_s = \prod_{i=1}^{n} E_{is} \tag{11}$$

for $s = 0, \ldots, t-1$.

The secret share distribution phase may introduce some additional communication overhead, but this phase only needs to be performed once.

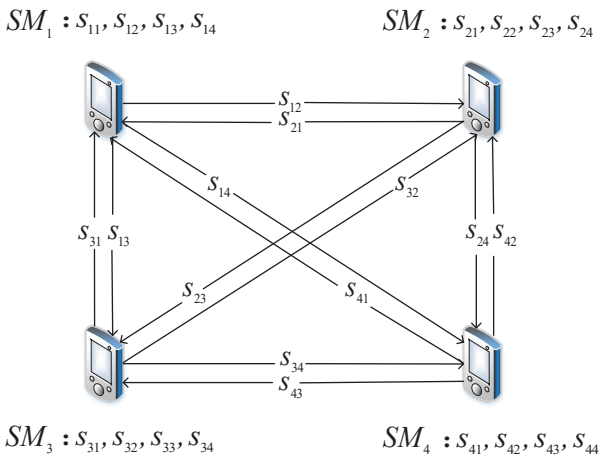
D. Data Reporting Stage

1. SM_i generates k types of electricity data $(m_{i1}, m_{i2}, \ldots, m_{ik})$, then computes $v_i = \sum_{j=1}^{k} m_{ij}\alpha_j$, where $m_{ij} \in [0, X]$, $[\log_2 nX] < k_1, X$ is the maximum value of electricity data.

2. SM_i randomly selects $y_i \in \mathbb{Z}_q^*$, and computes $Y_i = y_i \cdot P$, $\hat{Y}_i = y_i \cdot P_{pub}$. Then SM_i randomly selects $S_i \in \mathbb{Z}_q^*$, and computes ciphertext $c_i = v_i + S_i + H_1(\hat{Y}_i)$ mod q, $Ts_i = S_i - SS_i$ mod q, then publishes Ts_i and Y_i, where Mn can perform precomputation to reduce computational overhead. Mn computes $\hat{Y}_i = x \cdot Y_i$ and $\sum_{i=1}^{n} H_1(\hat{Y}_i)$.

3. SM_i randomly selects $r_i \in \mathbb{Z}_q^*$, and computes $R_i = r_i \cdot P$, $d_i = H_2(R_i, c_i, ID_i, T)$, $z_i = r_i + d_i \cdot x_i$, where T is the current timestamp and (R_i, z_i) is the signature of SM_i for ciphertext c_i.

4. Finally, SM_i sends (R_i, z_i, c_i, ID_i, T) to Mn.



After verifying the correctness of the received shares,

SM_1 computes secret share : $s_1 = s_{11} + s_{21} + s_{31} + s_{41}$

SM_2 computes secret share : $s_2 = s_{22} + s_{12} + s_{32} + s_{42}$

SM_3 computes secret share : $s_3 = s_{33} + s_{13} + s_{23} + s_{43}$

SM_4 computes secret share : $s_4 = s_{44} + s_{14} + s_{24} + s_{34}$

Fig. 3: Example for Secret Share Distribution (for convenience, (s_{ij}, t_{ij}) is replaced by s_{ij}).

E. Data Reading Stage

After receiving all user reports, Mn performs the following steps:

1. Check the identity ID_i and the timestamp T. If they are correct, compute $d_i = H_2(R_i, c_i, ID_i, T)$ and verify

$$z_i \cdot P = R_i + d_i \cdot PK_i \tag{12}$$

for $i = 1, \ldots, n$. Batch verification can be performed with small exponent test technology to increase speed. Mn randomly selects a group of small numbers $\theta_1, \theta_2, \ldots, \theta_n \in [1, 2^l]$ to verify if

$$\sum_{i=1}^{n} \theta_i z_i \cdot P = \sum_{i=1}^{n} \theta_i R_i + \sum_{i=1}^{n} \theta_i d_i \cdot PK_i \tag{13}$$

where l is a security parameter such that the probability of accepting a bad pair is 2^{-l}.

2. If the above verification is correct, compute the aggregation data

$$C = \sum_{i=1}^{n} c_i \bmod q \tag{14}$$

3. Use the collected t verifiable shares (s_j, t_j) to verify

$$E(s_j, t_j) = \prod_{s=0}^{t-1} E_s^{j^s} \tag{15}$$

to check the validity of the received secret shares. if they are all valid, according to the secret sharing homomorphism, Mn can get the sum $\sum_{i=1}^{n} SS_i$ of static secret of all users by Lagrange interpolation formula.

4. Compute

$$\sum_{i=1}^{n} S_i = \sum_{i=1}^{n} SS_i + \sum_{i=1}^{n} Ts_i \ c \tag{16}$$

$$\sum_{i=1}^{n} v_i = C - \sum_{i=1}^{n} S_i - \sum_{i=1}^{n} H_1(\hat{Y}_i) \mod q \tag{17}$$

holds for $\sum_{i=1}^{n} v_i < q$.

5. According to the Chinese Remainder Theorem, compute the sum of the data of all user in the j-th dimension $M = (M_1, M_2, \ldots, M_k)$.

$$M_j = (\sum_{i=1}^{n} v_i \bmod Q) \bmod q_j = \sum_{i=1}^{n} m_{ij} \tag{18}$$

Finally, Mn records the aggregation data into the new block. Mn broadcasts this block, and other nodes in the entire network link the block to their respective blockchain.. Then CC can query the block in the blockchain, and obtain the aggregated data M.

F. Dynamic User Management

1. The user join: Assuming a new user SM_r joins, following secret share distribution stage, SM_r randomly selects $S_r \in \mathbb{Z}_q^*$, and selects $n-1$ users to share secret, then other users update their own s_i after receiving the secret share of SM_r, so that Mn can collect at least t shares s_i' to recover S', Then Mn computes $V = \sum_{i=1}^{n} c_i + c_s - S' = \sum_{i=1}^{n} v_i + v_s$, and finally gets the aggregated multidimensional data.

2. The user leave: When a user SM_u leaves, it cannot send its secret value to Mn, so $V = \sum_{i \neq u}^{n} (v_i + S_i) - (\sum_{i=1}^{n} S_i) \neq \sum_{i \neq u}^{n} v_i$. To solve this problem, Mn can broadcast the message of SM_u, then all users update s_i to subtract the share from SM_u, so that Mn can collect at least t new s_i' to recover S', and compute $V = \sum_{i \neq u}^{n} c_i - S' = \sum_{i \neq u}^{n} v_i$.

G. Fault Tolerance

1. The user failure: Assuming some users' smart meters do not work, other users can perform the same operations as when the user left to recover the aggregated data of normal users.

2. The intermediate node malfunction: When the intermediate node Mn does not work, according to the Raft protocol, other smart meters will use the leader election algorithm to elect a new mining node to collect and aggregate data.

VI. Security Analysis

In this section, we carry out a strict security certification of our BBMDA scheme and analyze the secure requirements of the scheme.

A. Security Model

Definition 1: A BBMDA scheme is semantically secure against chosen plaintext attacks (IND-CPA) if no probabilistic polnomial time adversary A is able to win the game below with a non-negligible advantage.

Setup. The challenger C generates the system parametes and sends them to A.

Challenge. The attacker A outputs two messages of the same length m_0 and m_1. The challenger C randomly selects b, and computes the ciphertext c_b of message m_b, then sends it to A.

Guess. Finally, A outputs $b' \in \{0, 1\}$ as a guess for b. The advantage of the attacker A in the above game can be defined as

$$Adv_A^{IDN-CPA} = \left| Pr[b' = b] - \frac{1}{2} \right|$$

B. Security Certification

Theorem 1: If there is a PPT adversary A that can break the IND-CPA security of the BBMDA scheme with advantage $Adv_A^{IND-CPA}$, then there is a PPT algorithm B that can solve the ECCDH problem with advantage

$$Adv_B^{ECCDH} \geq Adv_A^{IND-CPA}$$

Proof: We present two games (**Game 0, Game 1**) as follows:

Game 0. This is the original IND-CPA game for our BMDA acheme.

1. The challenger C carries out system initialization to obtain public system parametes params= $(q, \mathbb{G}, P, H_1, P_{pub})$ and private key x, then C computes $P_{pub} = x \cdot P$ and sends (params, P_{pub}) to the adversary A.

2. The adversary A outputs two messages of the same length m_0 and m_1. The challenger C randomly selects b, computes the ciphertext $c_b = m_b + S + H_1(\hat{Y}) \bmod q$, and publishes Y, where

$$S, y \in \mathbb{Z}_q^*, \ Y = y \cdot P, \ \hat{Y} = y \cdot P_{pub}$$

Then C sends c_b to A.

3. At last, A outputs b' as a guess for b. If b' = b, A wins the game.

Game 1. The game is the same as **Game 0** except that the challenger C replaces $\hat{Y}$ with a random element R in $\mathbb{G}$.

Next, we analyze the above two games under the ECCDH assumption. we construct a distinguisher algorithm B and estimate its probability in distinguishing differences betwen **Game 0** and **Game 1**. Let E_i be the event that A wins the **Game i** (i.e. $1 \leftarrow C$) for $i = 0, 1$. By Definition1, the advantage of A in the **Game 0** can be defined as

$$Adv_A^{IND-CPA} = \left| Pr[E_0] - \frac{1}{2} \right| \tag{19}$$

■

Lema 1: If an adversary A can distinguish the difference between **Game 0** and **Game 1**, there is an algorithm B that can solve the ECCDH problem with the adavantage

$$Adv_B^{ECCDH} = Pr[E_0] - Pr[E_1] \tag{20}$$

Proof: With a given ECCDH instance $(\mathbb{G}, P, H_1, R)$ from its challenger, the algorithm B presents the following game:

1. The algorithm B randomly selects $x, y \in \mathbb{Z}_q^*$, and computes $P_{pub} = x \cdot P, Y = y \cdot P$.

Then B sends $(\mathbb{G}, P, P_{pub}, Y)$ to the adversary A.

2. the adversary A outputs two messages of the same length m_0 and m_1. The algorithm B randomly selects b, and computes the ciphertext $c_b = m_b + S + H_1(R)$ mod q.

Then B sends c_b to A.

3. At last, A outputs b' as a guess for b. If b' = b, A wins the game.

if $R = xyP$, the above game is exactly the same as the **Game 0**. Thus, we have

$$Pr[1 \leftarrow B|R = xyP] = Pr[E_0] \tag{21}$$

Otherwise, R is a random element in G, and the above game is exactly the same as the **Game 1**. Thus, we have

$$Pr[1 \leftarrow B|R \in G] = Pr[E_1] \tag{22}$$

Therefore, by Eq. (21) and (22), we directly have Eq. (20) and proof the lemma. ■

Lema 2: In **Game 1**, the adversary A has no advantage, i.e.

$$Pr[E_1] = \frac{1}{2} \tag{23}$$

Proof: In **Game 1**, the adversary A is given $c_b = m_b + S + H_1(R)$ mod q, where R is a random element, S is a random number, which are used only once. Therefore, A has no advantage of winning the game other than a random guess. ■

By combinning Eq. (19), (20) and Eq. (23), we can proof

$$Adv_A^{IND-CPA} \leq Adv_B^{ECCDH}$$

Since the ECCDH assumption says that Adv_B^{ECCDH} is negligible, we have $Adv_A^{IND-CPA}$ is negligible for all probabilistic polynomial time adversaries.

C. Security Requirements Analysis

1. Confidentiality

Scenario 1: It is not feasible to obtain the electricity consumption data of a single user from the ciphertext.

Proof: Firstly, since user ciphertext $c_i = m_i + S_i$ mod q, from Theorem1, we can know that it is not feasible to obtain m_i without knowing S_i. Secondly, the secret sharing scheme cannot recover the secret when there are fewer than t malicious users. Therefore, attackers cannot obtain the random number S_i, and it is not feasible to obtain individual electricity consumption data from the ciphertext. ■

2. Integrity

Scenario 2: BBMDA can ensure the data integrity of user data.

Proof: In our scheme, the secure Schnorr signature is used. In fact, the attacks who want to forge signatures either crack the hash function; or solve $ECDLP$. In this scheme, we use a secure hash function and the recommended elliptic curve, so the above-mentioned adversarial task is not feasible. Therefore, the data integrity of the user data is provided. ■

3. Privacy Preservation

Scenario 3: If the number of malicious users is fewer than t, no one can obtain the personal electricity data.

Proof: From Corollary 1, it can be seen that when there are at most t malicious users, the attacker cannot obtain the individual electricity data from the ciphertext. At the same time, since the secret sharing homomorphism scheme is adopted, the secret obtained is the sum of the secrets of all users. Therefore, Mn and CC can only obtain aggregated data M, and it is not feasible to derive individual power data from M. Therefore, no one, including internal attackers, can obtain individual electricity consumption data. ■

4. Identity Verification

Scenario 4: BBMDA can realize user identity authentication.

Proof: Firstly, illegal ID can be detected by Mn. Secondly, the secure Schnorr signature is employed to ensure the integrity of the user data, if an attacker tries to forge a legitimate user's ID to send wrong data, he cannot forge a legitimate user's signature and therefore it will be recognized by the system. ■

5. Forward Secrecy

Scenario 5: BBMDA can realize the forward secrecy of the user's security key.

Proof: Once in a while, all users perform the key update step. In this step, each user updates the dynamic secret S_i, and at the same time, the recovery parameter $Ts_i = S_i - SS_i$ is also updated. Therefore, even if the user's security key S_i is leaked, the user's previous ciphertext cannot be accessed. Therefore, forward secrecy is realized in our scheme. ■

6. Resistance Against Attacks

Scenario 5: BBMDA can resist modification attack, replay attack, impersonation attack, man-in-the-middle attack and internal attack.

Proof:

(1) Modification Attack : Scenario 2 proves that no attacker could forge a legal ciphertext. Mn can detect any modification of the received ciphertext verifying if $\sum_{i=1}^{n} \theta_i z_i \cdot P = \sum_{i=1}^{n} \theta_i R_i + \sum_{i=1}^{n} \theta_i d_i \cdot PK_i$. Therefore, our scheme can resist any modification attack.

(2) Replay Attack : The timestamp T is used in the message (R_i, z_i, c_i, ID_i, T) sent by SM to Mn, $d_i = H(R_i, c_i, ID_i, T)$, $z_i = r_i + d_i \cdot x_i$, so Mn could detect replay attack by verifying T's freshness. Therefore, our scheme can resist any replay attack.

(3) Man-in-the-Middle Attack : Scenario 4 proves that our scheme can realize user identity authentication. Mn can authenticate SM_i by checking if $z_i \cdot P = R_i + d_i \cdot PK_i$. Therefore, our scheme can resist any man-in-the-middle attack.

(4) Impersonation Attack : Scenario 4 proves that our scheme can realize user identity authentication. Therefore, our scheme can resist any impersonation attack.

(5) Internal Attack : Since the secret sharing homomorphism scheme is adopted, the secret obtained is the sum of the secrets of all users. Therefore, Mn and CC can only obtain aggregated data M, and it is not feasible to derive individual electricity

TABLE II Security Comparison with Related Schemes

Scheme	[9]	[10]	[13]	[15]	[16]	[20]	our
Confidentiality	Y	Y	Y	Y	Y	Y	Y
No Trusted Authority	Y	N	Y	Y	N	Y	Y
Privacy	Y	Y	Y	Y	Y	Y	Y
Dynamic User Management	Y	Y	Y	N	N	Y	Y
Multidimensional Data	Y	Y	N	N	N	Y	Y
Integrity	N	Y	N	N	Y	Y	Y
Authentication	N	Y	N	N	Y	Y	Y
Forward Secrecy	Y	Y	Y	N	Y	N	Y
Intermediate Node Fault Tolerance	N	N	N	N	N	N	Y

data from M. Therefore, our scheme can resist any internal attack. ■

D. Comparison of Security Features

We compare BBMDA with some excellent data aggregation schemes Wang [9], Mohammadali [10], Xue [14], Karampour [15], Liu [16], and Ming [20] in terms of the aspects of Confidentiality, No Trusted Authority, Privacy, Authentication, Integrity, Dynamic User Management, Multidimensional Data, Forward Secrecy, Intermediate Node Fault Tolerance. As shown in Table II, our scheme can meet all the above security requirements, thus showing stronger security than these existing works.

VII. Performance Evaluation

In this section, we explore the scheme performance in terms of the communication and computation overhead.

A. Computation Overhead

To evaluate the performance of our scheme, we compare BBMDA with the schemes Wang [9], Karampour [**?**], Liu [16], and Ming [20] in computation overhead. Yet, some lightweight operations (hash function and point addition) are not taken into account). The performance evaluation is executed in a computer with the Intel(R) Core(TM) i5-6500 CPU @ 3.20 GHz and 8 GB memory, using the MIRACL [26]. We use the standard curve secp160r1: $y^2 = x^3 + ax + b \bmod p$ with a prime order q, where p, q are 160 bits prime numbers and $a = -3, b$ is a random 160 bits prime number. In addition, we choose Tate pairing $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_{\mathbb{T}}$, which is implemented on a super-singular curve over $GF(p)$ with embedding degree 2. The runtime of the cryptographic operations needed in these schemes are listed in Table III.

In BBMDA, firstly, SM executes three scale multiplication operations in ECC. Secondly, Mn executes $n+1$ scale multiplication operations in ECC. Therefore, the runtime of our scheme is $(n+4) \cdot T_{m-ECC} = 0.31n + 1.24$ ms.

TABLE III Time Cost of Related Operation (Millisecond)

Notations	Description	Runtime
$\mathrm{T_{m-ECC}}$	Scale Multiplication Operation in ECC	0.31
$\mathrm{T_{log-ECC}}$	Solving the $ECDLP$ Operation	1.01
$\mathrm{T_b}$	Bilinear Pairing Operation	7.44
$\mathrm{T_{m-n^2}}$	Modular Multiplication in $\mathbb{Z}^*_{N^2}$	0.15
$\mathrm{T_{exp-q}}$	Modular Exponentiation in $\mathbb{Z}^*_q$	0.44
$\mathrm{T_{exp-n^2}}$	Modular Exponentiation in $\mathbb{Z}^*_{N^2}$	2.87

For the schemes Wang [9], Karampour [15], Liu [16],and Ming [20], we follow the same procedure to compute the computation overhead of all entities as shown in IV.

TABLE IV Comparison of Computation Overhead (ms)

Scheme	SM	Mn	CC	Toal Cost
Wang [9]	$8T_{m-ECC} + 3T_b + 4T_{exp-n^2}$	nT_{m-ECC}	$3T_{m-n^2} + T_{exp-n^2}$	$0.31n + 39.6$
Karampour [15]	$nT_{m-n^2+2T_{exp-n^2}}$	nT_{m-n^2}	$3T_{m-n^2} + T_{exp-n^2}$	$3.17n + 9.06$
Liu [16]	$8T_{m-ECC} + T_b$	$(2n+3)T_{m-ECC} + 3T_b$	$2T_b$	$0.62n + 44.95$
Ming [20]	$4T_{m-ECC}$	$(n+2)T_{m-ECC}$	$4T_{m-ECC} + T_{log-ECC}$	0.31n+4.11
Our	$3T_{m-ECC}$	$(n+1)T_{m-ECC}$		$0.31n + 1.24$

Fig.4 and Fig.5 show the comparison of computation overhead between our scheme and the other schemes above. In Fig.5, the number of users is assumed as 100 and the data type is 16. It can be seen from Fig.4 and Fig.5 that our scheme is more efficient, and our scheme and Ming [20] can aggregate multidimensional data to reduce computation overhead.

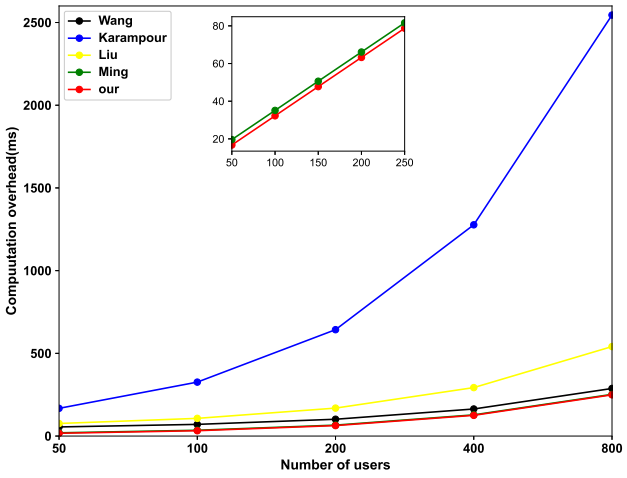


Fig. 4: Computation Overhead vs. Number of Users.

B. Communication Overhead

In smart grid, the communication overhead is generated as a result of the communication between all entities. We divide

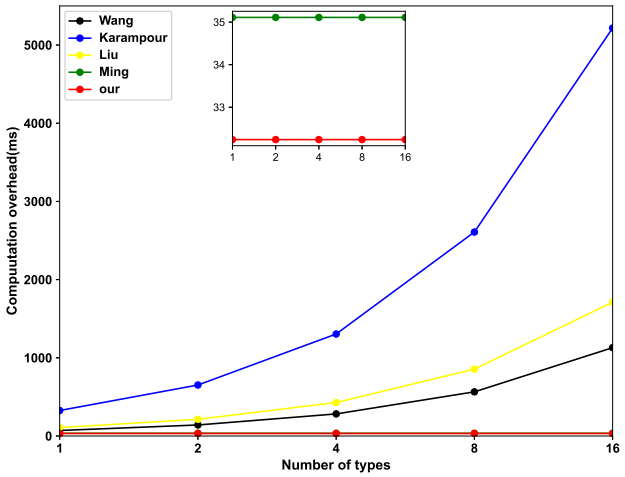


Fig. 5: Computation Overhead vs. Number of Data Types.

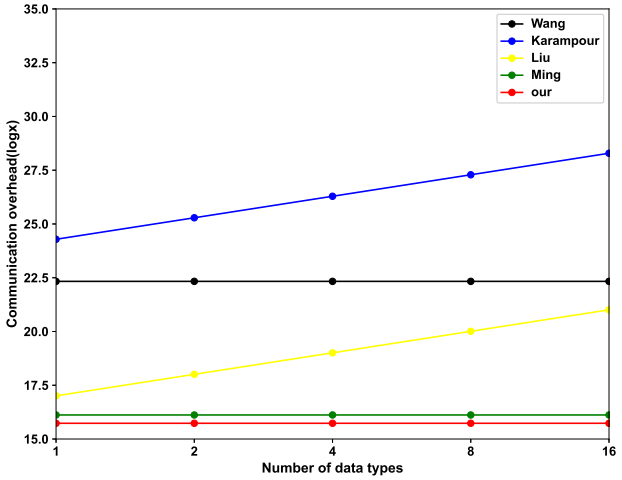


Fig. 6: Communication Overhead vs. Number of Users.

the communication overhead into three parts: SM-SM, SM-Mn and Mn-CC. In BBMDA , the message transmission from SM to SM only needs to be carried out once, so it is not considered. We assume that the sizes of elements in $\mathbb{G}, \mathbb{Z}_q^*, \mathbb{Z}_n^*, \mathbb{Z}_{\hat{p}}^*, \mathbb{Z}_{\hat{q}}^*, \mathbb{Z}_{n^2}$ are 160 bits, 160 bits, 1024 bits, 1024 bits, 160 bits and 2048 bits. Both sizes of the timestamp and the identity are 32 bits. The communication overhead in each phase of schemes Wang [9], Karampour [15], Liu [16], Ming [20] and BBMDA are shown in Table V.

TABLE V Comparison of Communication Overhead (bit)

Scheme	SM-SM	SM-Mn	Mn-CC
Wang [9]	$512n(n-1)$	$2048n$	2048
Karampour [15]	$n(2048(n-1))$	$2048n$	2048
Liu [16]		$928n$	$384n+544$
Ming [20]		$704n$	704
Our		$544n$	

In BBMDA, for SM-Mn, each SM_i sends message (R_i, z_i, c_i, ID_i, T) to Mn, where $R_i \in \mathbb{G}$, z_i, $c_i \in \mathbb{Z}_q^*$, ID_i is a 32-bit identity and T is a 32-bit timestamp. Therefore, the communication overhead is 160 + 160 + 160 + 32 + 32 = 544 bits. For Mn-CC, since Mn is not required to send data directly to CC in BBMDA, the communication cost is 0 bit.

The comparison of the communication overhead is shown in Fig.6 and Fig.7. In Fig.7, the number of users is assumed as 100. For convenience, the communication overhead in Fig.6 and Fig.7 is the logarithm of the real data with base 2.

It can be seen from Fig.4, Fig.5, Fig.6 and Fig.7 that our scheme has lower computation and communication overhead than other schemes. Therefore, our scheme is more suitable for smart grid.

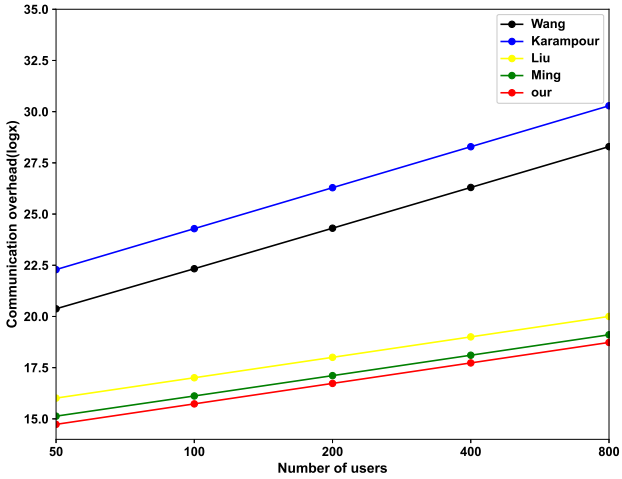


Fig. 7: Communication Overhead vs. Number of Data Types.

VIII. Conclusion

In this paper, we propose an efficient and robust data aggregation scheme based on blockchain. Our scheme does not need any trusted authority, and it can realize the multi-dimensional data aggregation based on the Chinese Remainder Theorem and support flexible dynamic user management and fault tolerance. Our security analysis has shown that our proposed scheme can meet stronger security features. Moreover, our performance evaluation proves that our scheme is more effective than some referenced works. In the future, we will improve our work to resist collusion attacks with k users.

Acknowledgment

This research is partially supported by the National Natural Science Foundation of China (No. 61772166) and the Key Program of the Natural Science Foundation of Zhejiang Province of China (No. LZ17F020002).

REFERENCES

[1] N. Komninos, E. Philippou, and A. Pitsillides, "Survey in smart grid and smart home security: Issues, challenges and countermeasures," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1933–1954, 2014.

[2] G. Si, Z. Guan, J. Li, P. Liu, and H. Yao, "A comprehensive survey of privacy-preserving in smart grid," in *International Conference on Security, Privacy and Anonymity in Computation, Communication and Storage*. Springer, 2016, pp. 213–223.

[3] L. Chen, R. Lu, and Z. Cao, "Pdaft: A privacy-preserving data aggregation scheme with fault tolerance for smart grid communications," *Peer-to-Peer networking and applications*, vol. 8, no. 6, pp. 1122–1132, 2015.

[4] O. R. Merad-Boudia and S. M. Senouci, "An efficient and secure multidimensional data aggregation for fog-computing-based smart grid," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6143–6153, 2020.

[5] J. Ni, K. Zhang, K. Alharbi, X. Lin, N. Zhang, and X. S. Shen, "Differentially private smart metering with fault tolerance and range-based filtering," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2483–2493, 2017.

[6] S. Ge, P. Zeng, R. Lu, and K.-K. R. Choo, "Fgda: Fine-grained data analysis in privacy-preserving smart grid communications," *Peer-to-Peer Networking and Applications*, vol. 11, no. 5, pp. 966–978, 2018.

[7] S. Li, K. Xue, Q. Yang, and P. Hong, "Ppma: Privacy-preserving multisubset data aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 462–471, 2017.

[8] F. Knirsch, G. Eibl, and D. Engel, "Error-resilient masking approaches for privacy preserving data aggregation," *IEEE Transactions on Smart Grid*, vol. 9, no. 4, pp. 3351–3361, 2016.

[9] X. Wang, Y. Liu, and K.-K. R. Choo, "Fault-tolerant multisubset aggregation scheme for smart grid," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 6, pp. 4065–4072, 2020.

[10] A. Mohammadali and M. S. Haghighi, "A privacy-preserving homomorphic scheme with multiple dimensions and fault tolerance for metering data aggregation in smart grid," *IEEE Transactions on Smart Grid*, vol. 12, no. 6, pp. 5212–5220, 2021.

[11] H. Fan, Y. Liu, and Z. Zeng, "Decentralized privacy-preserving data aggregation scheme for smart grid based on blockchain," *Sensors*, vol. 20, no. 18, pp. 1–14, 2020.

[12] A. Alsharif, M. Nabil, S. Tonyali, H. Mohammed, M. Mahmoud, and K. Akkaya, "Epic: Efficient privacy-preserving scheme with etoe data integrity and authenticity for ami networks," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 3309–3321, 2018.

[13] Z. Guan, G. Si, X. Zhang, L. Wu, N. Guizani, X. Du, and Y. Ma, "Privacy-preserving and efficient aggregation based on blockchain for power grid communications in smart communities," *IEEE Communications Magazine*, vol. 56, no. 7, pp. 82–88, 2018.

[14] K. Xue, B. Zhu, Q. Yang, D. S. Wei, and M. Guizani, "An efficient and robust data aggregation scheme without a trusted authority for smart grid," *IEEE Internet of Things Journal*, vol. 7, no. 3, pp. 1949–1959, 2019.

[15] A. Karampour, M. Ashouri-Talouki, and B. T. Ladani, "An efficient privacy-preserving data aggregation scheme in smart grid," in *2019 27th Iranian Conference on Electrical Engineering (ICEE)*. IEEE, 2019, pp. 1967–1971.

[16] Y. Liu, W. Guo, C.-I. Fan, L. Chang, and C. Cheng, "A practical privacy-preserving data aggregation (3pda) scheme for smart grid," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1767–1774, 2018.

[17] H. Shen, Y. Liu, Z. Xia, and M. Zhang, "An efficient aggregation scheme resisting on malicious data mining attacks for smart grid," *Information Sciences*, vol. 526, pp. 289–300, 2020.

[18] H. Shen, M. Zhang, and J. Shen, "Efficient privacy-preserving cube-data aggregation scheme for smart grids," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1369–1381, 2017.

[19] Y. Chen, J.-F. Martínez-Ortega, P. Castillejo, and L. López, "A homomorphic-based multiple data aggregation scheme for smart grid," *IEEE Sensors Journal*, vol. 19, no. 10, pp. 3921–3929, 2019.

[20] Y. Ming, X. Zhang, and X. Shen, "Efficient privacy-preserving multi-dimensional data aggregation scheme in smart grid," *IEEE Access*, vol. 7, pp. 32 907–32 921, 2019.

[21] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized Business Review*, pp. 1–9, 2008.

[22] S. Cho and S. Lee, "Survey on the application of blockchain to iot," in *2019 International Conference on Electronics, Information, and Communication (ICEIC)*. IEEE, 2019, pp. 1–2.

[23] D. Ongaro and J. Ousterhout, "In search of an understandable consensus algorithm," in *2014 {USENIX} Annual Technical Conference ({USENIX}{ATC} 14)*, 2014, pp. 305–319.

[24] T. P. Pedersen, "Non-interactive and information-theoretic secure verifiable secret sharing," in *Annual international cryptology conference*. Springer, 1991, pp. 129–140.

[25] J. C. Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret secret," in *Conference on the theory and application of cryptographic techniques*. Springer, 1986, pp. 251–260.

[26] M.Scott, *Multiprecision Integer and Rational Arithmetic C/C++ Library (MIRACL)*, [online] Available : https://github.com/miracl/MIRACL, Apr. 2016.