applied sciences

MDPI

Article

Privacy-Preserving Data Aggregation with Dynamic Billing in Fog-Based Smart Grid

Huiyong Wang [1,2], Yunmei Gong [1], Yong Ding [2,3], Shijie Tang [4,*] and Yujue Wang [5]

[1] School of Mathematics and Computing Science, Guilin University of Electronic Technology, Guilin 541004, China
[2] Guangxi Key Laboratory of Cryptography and Information Security, School of Computer Science and Information Security, Guilin University of Electronic Technology, Guilin 541004, China
[3] Cyberspace Security Research Center, Peng Cheng Laboratory, Shenzhen 518055, China
[4] School of Electronic Engineering and Automation, Guilin University of Electronic Technology, Guilin 541004, China
[5] Hangzhou Innovation Institute, Beihang University, Hangzhou 310052, China
* Correspondence: tangsj@guet.edu.cn

Abstract: As the next-generation grid, the smart grid (SG) can significantly enhance the reliability, flexibility as well as efficiency of electricity services. To address latency and bandwidth issues during data analysis, there have been attempts to introduce fog computing (FC) in SG. However, fog computing-based smart grid (FCSG) face serious challenges in security and privacy. In this paper, we propose a privacy-preserving data aggregation scheme that supports dynamic billing and arbitration, named PPDB. Specifically, we design a four-layer data aggregation framework which uses fog nodes (FNs) to collect and aggregate electricity consumption data encrypted under the ElGamal cryptosystem and employ distributed decryption to achieve fine-grained access and bills generation based on real-time prices. In addition, we introduce a trusted third party to arbitrate disputed bills. Detailed security analysis proves that the proposed PPDB can guarantee the confidentiality, authentication and integrity of data. Compared with related schemes, the experimental results show that the communication overhead of our scheme is reduced by at least 38%, and the computational efficiency in the billing phase is improved by at least 40 times.

Keywords: smart grid; privacy-preserving; fog computing; data aggregation; dynamic billing; arbitration

Citation: Wang, H.; Gong, Y.; Ding, Y.; Tang, S.; Wang, Y. Privacy-Preserving Data Aggregation with Dynamic Billing in Fog-Based Smart Grid. *Appl. Sci.* **2023**, *13*, 748. https://doi.org/10.3390/app13020748

Academic Editor: Giacomo Fiumara

Received: 21 November 2022
Revised: 24 December 2022
Accepted: 29 December 2022
Published: 5 January 2023

1. Introduction

As one of the typical application scenarios in smart cities [1], the smart grid (SG) aims to solve the problems of low reliability, frequent power outages and high carbon emissions of traditional power grids by integrating various modern emerging technologies [2,3]. Smart meters (SMs) are the core component of the SG. They can generate real-time electricity consumption data of users and periodically (e.g., every 15 min [4]) report the data to the service provider (SP) [5], who can analyse the data to anticipate electricity demands and adjust electricity generation and distribution [6]. Such a mechanism can significantly improve the reliability, flexibility and efficiency of the power system, thereby providing higher-quality services to users [7].

Since the number of SMs in grid-covered areas is exploding [8], and the real-time electricity consumption data generated by SMs have the characteristics of being high volume, velocity as well as variety [9], transmitting large amounts of data directly to the SP imposes heavy computational and communication-related burden on the system. Meanwhile, the SP may not be able to process the massive amount of data in time [10], thus inducing network latency, which is impractical for delay-sensitive SG applications. To address latency and bandwidth issues during data analysis, there have been attempts to introduce fog computing (FC) into the SG [5,6,10–14].

FC has attracted great attention since CISCO [15] proposed the concept in 2012. As a supplement to the cloud computing model, FC enables users to realize computation, communication and storage locally by extending computing power to the network edge [16]. Moreover, FC has advantages over cloud computing, such as low latency and location awareness [17], which satisfies the delay-sensitive demand of SG applications and can improve the real-time performance and quality of service of the SG system [8]. Considering the promotion of FC for the sustainable development of SG, introducing FC into the SG is a current hot topic. In fog computing-based smart grid (FCSG), fog nodes (FNs) deployed at the edge of the network are responsible for collecting all users' electricity consumption data in their coverage area, then aggregating and forwarding these data to the SP [5]. This type of architecture relieves much of the burden on the SP [18] and achieves lower network latency and higher bandwidth.

Although the FCSG is promising, it still faces serious challenges in terms of security and privacy [12], such as learned personal patterns, energy theft and impersonation [19], etc. Specifically, adversaries can infer the lifestyles and behavioural habits of users by analysing their real-time electricity consumption data to carry out some improper activities [20]. Furthermore, adversaries can also tamper with and forge users' electricity consumption data, thus threatening the stability of the SG [13]. For instance, network attacks launched by malicious adversaries caused the most significant blackout of electricity in Ukrainian history in 2015 [21]. Thus, protecting electricity consumption data has great practical importance.

How to improve efficiency and performance while ensuring the security of electricity consumption data has attracted a lot of attention from both academia and industry, and many privacy-preserving schemes have been proposed. However, existing privacy-preserving schemes rarely consider dynamic billing. In the traditional grid, SP generally charges users based on fixed prices [22]. However, such billing methods cannot reflect the relationship between electricity supply and demand, and the SP is thus unable to make timely adjustments when electricity demand fluctuates, resulting in a huge waste of resources [23]. With the rapid development of SG, a real-time-prices-based billing strategy is proposed, which can guide users to adjust their consumption patterns according to dynamically changing prices and improve the efficiency of resource utilization [24]. Currently, many countries (such as Sweden, Norway, Portugal, etc.) have already adopt real-time-prices-based tariffs to charge their users [25]. Clearly, it is worthwhile to investigate how to construct a system with strong security and efficiency and which supports dynamic billing.

1.1. Related Works

Homomorphic encryption techniques are widely employed in privacy-preserving data aggregation schemes due to their homomorphic properties. In this article, we briefly review some data aggregation articles based on homomorphic encryption.

In traditional homomorphic encryption-based data aggregation schemes, SMs send reports to a gateway node, then the gateway node aggregates all reports and transmits the results to the SP, thus allowing the SP to obtain total electricity consumption data [6]. Lu et al. [4] proposed a multi-dimensional data aggregation scheme in which they used a super-increasing sequence to construct multi-dimensional data and encrypted the structured data using the Paillier cryptosystem. Some researchers constructed multi-subset data aggregation schemes [26,27] using super-increasing sequences and homomorphic encryption techniques. In multi-subset data aggregation schemes, the SP is able to calculate the total electricity consumption of all users and the number of users in different electricity consumption ranges. However, in [27], the computation overhead at the SM side is high, which is impractical for SMs which have limited computational power. Xue et al. [28] utilized the Paillier cryptosystem and secret sharing techniques to achieve privacy protection and flexible management of users. However, their scheme has high computation overhead since the keys need to be updated at each time slot. Shen et al. [29] proposed a new type of attack, namely, the malicious data mining attack. Considering this attack, they proposed a privacy-preserving data aggregation scheme based on the Paillier cryptosystem.

Compared with traditional data aggregation architectures, FC-based aggregation architectures can reduce latency and improve bandwidth utilization in time-sensitive applications [30]. So far, researchers have proposed a number of data aggregation schemes based on FC. Boudia et al. [6] came up with a multi-dimensional data aggregation scheme. They utilized binary encoding functions to construct multi-dimensional data into one-dimensional data and encrypt the constructed data using the Paillier cryptosystem. Compared with the super-increasing sequence mechanism, their scheme avoids the use of large coefficients and significantly reduces the ciphertext space. Liu et al. [10] suggested a scheme that focuses on both communication aggregation and function query. They utilized the double trapdoor cryptosystem to encrypt data and outsourced the ciphertext to the cloud, so users and the SP can initiate function queries on data in the cloud while protecting privacy. In [11], Liu et al. proposed a data aggregation scheme supporting multi-party computation in which the SP and users can make functional computation queries on the data. Khan et al. [12] achieved privacy protection of data by using the Boneh–Goh–Nissam cryptosystem and authenticated the source of data via the Elliptic Curve Digital Signature Algorithm. Singh et al. [31] submitted a multi-dimensional data aggregation scheme based on fog–cloud architecture, named PP-MDA. In PP-MDA, they employed two outsourced clouds to reduce the computation and storage overhead. Zhang et al. [32] added blinding factors to the encrypted data, so in their scheme, the adversary cannot recover individual data even if the symmetric key is compromised.

1.2. Motivations and Contributions

Although some FC-based privacy-preserving schemes have been proposed, most of them only focus on privacy-preserving data aggregation, while few schemes implement privacy-preserving dynamic billing. Furthermore, in the existing cloud-based dynamic billing article [23,25,33], the SP can only calculate the total electricity bill of the user in one billing cycle but cannot calculate the electricity bill generated by different types of electrical appliances. In addition, to our knowledge, there is no current work to arbitrate disputed bills. Based on the above analysis, our goal is to propose an FC-based privacy-preserving data scheme that supports dynamic billing and arbitration. Specifically, it provides the following functionalities:

- **Privacy-preserving aggregation:** In PPDB, each SM periodically generates electricity usage data for l types of appliances. Inspired by the scheme [6], we adopt a binary encoding function to construct the l-dimensional data, then encrypt the constructed data using an additive homomorphic variant of the ElGamal cryptosystem [34] and finally send the ciphertext to the FN. The FN then employs batch verification algorithms to check the validity of all the ciphertexts and sends the valid ciphertexts to the next level after aggregation, thus effectively saving on computation and communication overhead.
- **Secure dynamic billing:** Our PPDB supports secure billing based on real-time prices, which means that our solution can provide each user with their electricity bill based on prices that change over time and also prevent privacy leaks to other sides. It is worth noting that in PPDB, the SP is not only able to calculate the total electricity bill of the user in a billing cycle, but also to calculate the cost generated by each type of appliance, respectively. Users can dynamically adjust their energy use patterns in response to price changes, thus optimizing resource utilization.
- **Dispute arbitration:** We also introduced a trusted third party as an arbitration centre (AC) to arbitrate disputes between SP and users over their electricity bills. To our knowledge, this is the first scheme that considers the arbitration of billing disputes.

The security of our scheme is analysed under the Decisional Diffie–Hellman (DDH) assumption and Computational Diffie–Hellman (CDH) assumption, which implies that under the chosen plaintext attack, our scheme can provide strong confidentiality, and under chosen message attacks, the adversary cannot forge a valid report. For evaluating the performance of the scheme, we conduct in-depth theoretical analysis and experimen-

tal comparison. The results show that the proposed PPDB has low computation and communication overheads.

1.3. Paper Organization

The remainder of this paper is organized as follows. We review some concepts related to our scheme in Section 2. Models and security requirements are described in Section 3. Our scheme is formalized in Section 4. The soundness and security analysis of the proposed scheme are presented in Section 5. The comparison and evaluation of the scheme are given in Section 6. In the end, this paper is summarised in Section 7.

2. Preliminaries

In this section, we introduce some preliminaries associated with the construction and security analysis of our scheme.

2.1. Bilinear Maps

Let $G = \langle g \rangle$ and G_τ be two cyclic groups of prime order q. We say that the mapping e: $G \times G \to G_\tau$ is bilinear if the following properties are satisfied:

- Bilinearity: $\forall g_1, g_2 \in G$ and $\forall a, b \in Z_q^*, e({g_1}^a, {g_2}^b) = e(g_1, g_2)^{ab}$;
- Non-degeneracy: $\exists g_1, g_2 \in G, e(g_1, g_2) \neq 1_{G_\tau}$;
- Computability: $\forall g_1, g_2 \in G$, there exists valid algorithms to compute $e(g_1, g_2)$ in polynomial time.

2.2. An Additive Homomorphic Variant of the ElGamal Cryptosystem

The additive ElGamal cryptosystem is a variant of the ElGamal cryptosystem, which contains the following three algorithms.

- Key generation: Suppose $Z_q^* = \langle g \rangle$ is a cyclic group of prime order q. Choose an element $x \in Z_q^*$ at random and calculate $y = g^x$. Set the public key as (y, g, q), and the corresponding secret key as x .
- Encryption: For a given message m, randomly select an element $r \in Z_q^*$ and compute the ciphertext as $C = E(m) = (u, v) = (g^r, y^r g^m)$.
- Decryption: For a given ciphertext C, use the private key to compute $g^m = \frac{v}{(u)^x}$. Finally, apply the Pollard lambda method [35] to obtain m.

The additive ElGamal cryptosystem has additive homomorphic properties and constant multiplicative homomorphic properties [34], i.e., for given ciphertexts $E(m_1)$ and $E(m_2)$, we have $E(m_1 + m_2) = E(m_1) \oplus E(m_2)$, $E(m)^n = (u, v)^n = (g^{nr}, y^{nr} g^{nm})$, where n is a constant.

2.3. Complexity Assumptions

- DDH assumption: Given a cyclic group $G = \langle g \rangle$ of prime order q. For any probability polynomial-time (PPT) adversary $\mathcal{A}$, given a tuple (q, g, g^x, g^y) for random elements $x, y \in Z_q^*$, the advantage for $\left| \Pr[\mathcal{A}(g, q, g^x, g^y, h_b] = b - \frac{1}{2} \right|$ is negligible, where $b \in \{0, 1\}, h_0 = g^z, h_1 = g^{ab}, z \in Z_q^*$.
- CDH assumption: Given a cyclic group $G = \langle g \rangle$ of prime order q. For any PPT adversary $\mathcal{A}$, given a tuple (q, g, g^x, g^y) for random elements $x, y \in Z_q^*$, the advantage of computing g^{xy} is negligible.

In order to protect the privacy of electricity consumption data m, we employ an additive homomorphic variant of the ElGamal cryptosystem (under the DDH assumption, the ElGamal encryption scheme is semantic secure [36]) to encrypt it and generate the ciphertext (C_1, C_2). Before sending the (C_1, C_2) to the next level, in order to enable the receiver to check whether the (C_1, C_2) has been tampered with or forged during public channel transmission, we use the BLS signature algorithm (under the CDH assumption, the BLS signature scheme was proven to be secure [37]) to generate the signature σ of (C_1, C_2),

and send (C_1, C_2, σ) to the receiver. According to the structure of the BLS signature, after receiving (C_1, C_2, σ), the receiver uses bilinear maps to check whether it is valid. The relationship between Sections 2.1–2.3 is shown in Figure 1.

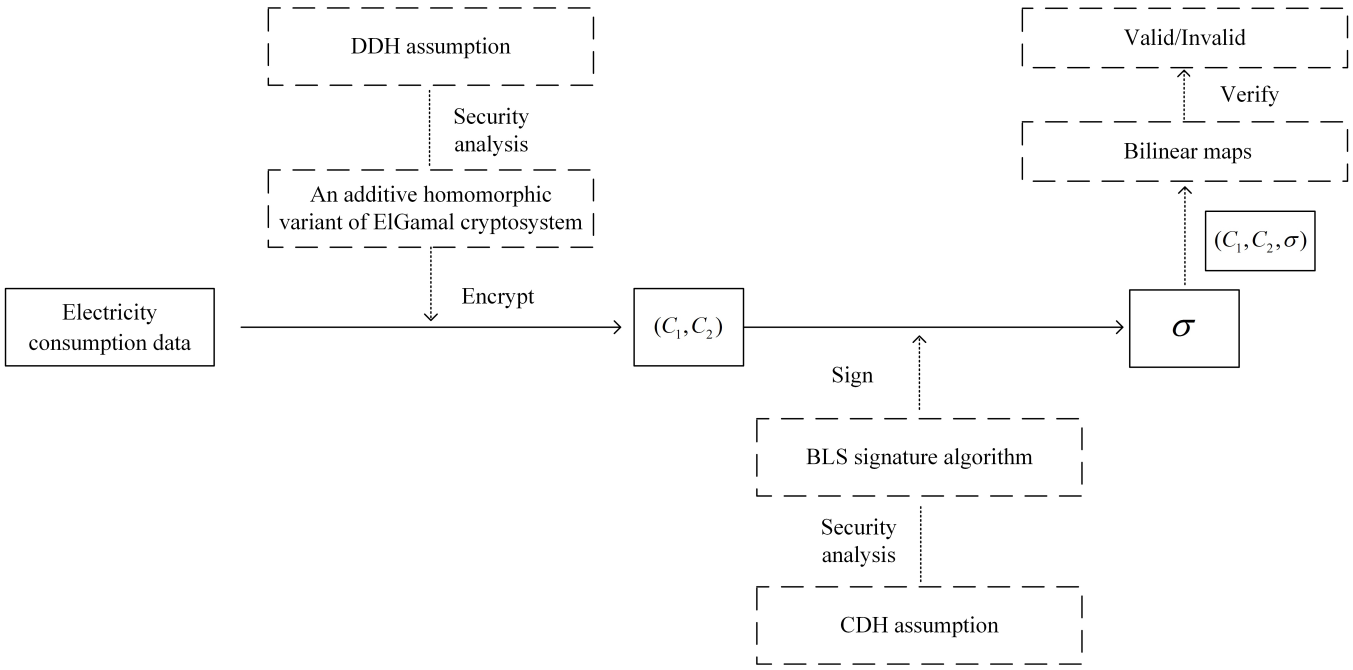


Figure 1. Relationship between Sections 2.1–2.3.

3. Problem Formalization

In this section, we detail the system model, threat model, security requirements and formal security definitions of PPDB.

3.1. System Model

In order to show the responsibilities and information interaction of each entity in PPDB more clearly, we present a system model. As shown in Figure 2, the system model includes five types of entities, namely, root key generation centre (RKG), SP, AC, FNs and SMs. We assume that the area covered by the grid is divided into k sub-regions. There are n SMs in each sub-region, and these n SMs are all covered by the same FN. $SM_{ij}(i = 1, 2, \cdots, k, j = 1, 2, \cdots, n)$ represents the j th SM in the i th sub-region, and $FN_i(i = 1, 2, \cdots, k)$ denotes the FN covering all SMs in the i th sub-region.

- *SMs:* SMs generate users' real-time electricity consumption data and encrypt data, then send the ciphertext to the corresponding FN periodically.
- *FNs:* After receiving reports from all SMs in the coverage area, FNs first verify the validity of all reports and then forward the valid reports to AC after aggregation. In addition, FNs can store the reports for billing.
- *AC:* Following receipt of the reports from FNs, AC validates their legitimacy, then pre-decrypts the reports and finally sends results to the SP. Furthermore, AC is tasked with the arbitration of disputes between users and the SP.
- *SP:* The SP is responsible for collecting total electricity consumption data, and charging users based on real-time prices.
- *RKG:* RKG takes charge of generating the public parameters of the system and generating keys for each entity.

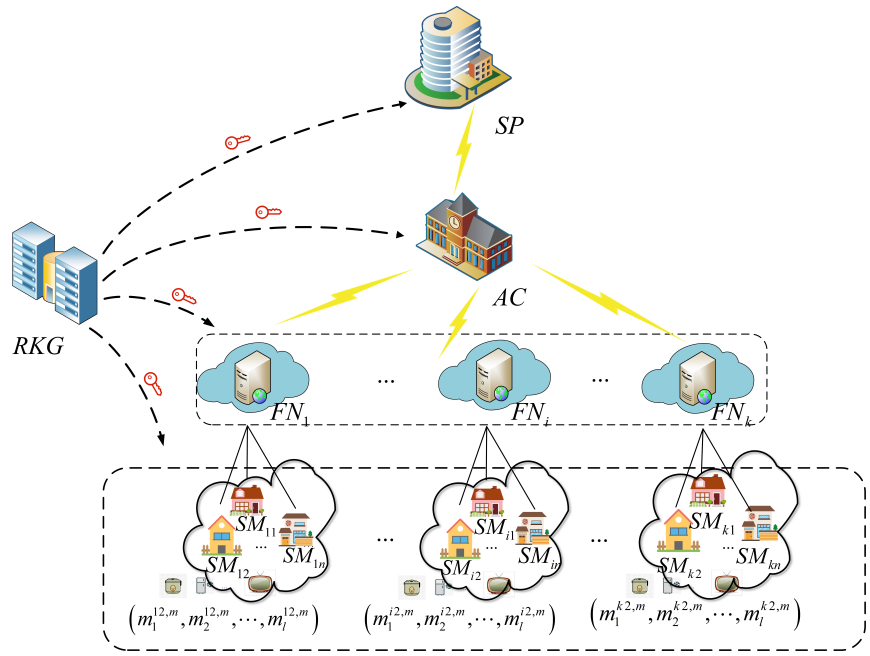


Figure 2. System model.

3.2. Threat Model

To demonstrate the security and feasibility of PPDB, we considered threats faced by SGs in the real world. In this paper, AC and RKG are considered to be honest, and they always strictly execute the protocol. SMs, FNs and SP are considered to be honest-but-curious, which means that they do not tamper with data but attempt to infer private information about the others. We define an honest-but-curious entity as an internal PPT adversary. In addition, we also consider the threats from an external PPT adversary $\mathcal{A}$. We assume that $\mathcal{A}$ has the following attack capabilities:

- $\mathcal{A}$ can eavesdrop on reports transmitted through public communication channels.
- $\mathcal{A}$ can intrude into the databases of FN_s, AC and SP to steal reports.
- $\mathcal{A}$ can also launch a number of active attacks to compromise the data authentication and integrity.

3.3. Security Requirements

Based on the proposed threat model, we defined the following security requirements. When the proposed PPDB satisfies the security requirements, it indicates that PPDB is able to resist the attacks mentioned in the threat model.

- **Confidentiality:** Reports transmitted through public communication channels and stored in databases may contain users' privacy information. Thus, all reports should be sent and stored in ciphertext format. Neither external nor internal adversaries can reveal the contents of the ciphertext even if they can obtain it; thus, the confidentiality of the data is preserved.
- **Authentication and Integrity:** $\mathcal{A}$ may impersonate legal entities to send false reports or tamper with them during transmission. Hence, all reports sent by legitimate entities should be authenticated, and any tampering with them should be detected.

3.4. Formal Security Definitions

By defining two games, we reduce the security of PPDB to difficult assumptions, that is, if PPDP is not secure, the difficult assumptions can be solved.

For the confidentiality of the data in our scheme, we define the following game.

Setup: The challenger $\mathcal{C}$ first generates the system parameters *param* and private key *sk*, and sends *param* to the adversary $\mathcal{A}$.

Challenge: The adversary $\mathcal{A}$ chooses two pieces of data m_0 and m_1, and sends them to the challenger $\mathcal{C}$, where $|m_0| = |m_1|$. The challenger $\mathcal{C}$ randomly chooses $b \in \{0,1\}$, computes $C^* = ENC(param, m_b)$ and sends it to the adversary $\mathcal{A}$.

Output: The adversary $\mathcal{A}$ outputs a guess b'. If $b' = b$, then the adversary $\mathcal{A}$ wins the game.

The advantage of the adversary $\mathcal{A}$ to win the game can be defined as $Adv_{\varepsilon,\mathcal{A}}(l) = \left| pr(b = b') - \frac{1}{2} \right|$.

Definition 1. *We say that the scheme enjoys indistinguishability under chosen plaintext attacks (IND-CPA) if any PPT adversary $\mathcal{A}$ has only negligible advantage in ε in winning the above game.*

We continue to define the EU-CMA security of scheme.

Setup: The challenger $\mathcal{C}$ first generates the system parameters *param* and private key *sk* and sends *param* to the adversary $\mathcal{A}$.

Signature Query: The adversary $\mathcal{A}$ adaptively selects data m_i and sends them to the challenger $\mathcal{C}$. The challenger $\mathcal{C}$ executes the signature algorithm to generate a signature σ_{m_i} and sends it to the adversary $\mathcal{A}$.

Output: The adversary $\mathcal{A}$ outputs a forged signature σ_{m^*} for a message m^*, where m^* has never been executed as a signature query before. If σ_{m^*} is a valid signature for m^*, then the adversary $\mathcal{A}$ wins the game.

Definition 2. *We say that the scheme enjoys existential unforgeability against chosen message attack (EU-CMA), if any PPT adversary $\mathcal{A}$ has only negligible advantage in ε in winning the above game.*

4. Our Proposal

In this section, we present a concrete privacy-preserving data aggregation scheme in the context of FCSG, which supports dynamic billing and arbitration. Table 1 describes the frequently used notations.

Table 1. Notation.

Symbol	Description
G, G_τ	Cyclic groups of prime order q
g	A generator of G
e	Bilinear map e: $G \times G \to G_\tau$
H	Hash function $H : \{0,1\}^* \to G$
$ID_{SM_{ij}}$	The identity of SM_{ij}
ID_{FN_i}	The identity of FN_i
ID_{AC}	The identity of AC
ID_{SP}	The identity of SP
x_{ij}, pk_{ij}	Private key, public key of SM_{ij}
$\sigma_{ij,m}$	Signature of SM_{ij} in time slot m
x_{FN_i}, pk_{FN_i}	Private key, public key of FN_i
$\sigma_{FN_{i,m}}$	Signature of FN_i in time slot m
$\sigma_{FN'_{ij}}$	Signature of FN_i at the billing stage
x_{AC}, pk_{AC}	Private key, public key of AC
$\sigma_{AC_{i,m}}$	Signature of AC in time slot m
$\sigma_{AC'_{ij}}$	Signature of AC at the billing stage
x_{SP}, pk_{SP}	Private key, public key of SP
σ_{SP}	Signature of SP
p_m	The electricity price in time slot m
$\|\|$	String concatenation operation

4.1. Scheme Overview

The proposed PPDB consists of seven phases: **System Initialization, Registration, Reports Generation, Data Aggregation, Data Reading, Billing and Arbitration**. The overview flow is as follows:

- **System Initialization:** PKG boots the entire system and generates system parameters.
- **Registration:** PKG generates and distributes keys for the entity requesting registration.
- **Reports Generation:** SM first encrypts the electricity consumption data, then sends the ciphertext to the corresponding FN.
- **Data Aggregation:** FN aggregates the valid ciphertext and then sends the aggregated ciphertext to AC.
- **Data Reading:** AC first pre-decrypts the received data, then sends the pre-decrypted ciphertext to SP. SP recovers the sum of electricity consumption data from the pre-decrypted data.
- **Billing:** FN computes the user's electricity billing ciphertext and sends the aggregated billing ciphertext to AC at the end of a billing cycle. AC first pre-decrypts the received billing ciphertext, then sends the pre-decrypted ciphertext to the SP. The SP decrypts the ciphertext to obtain each user's bill.
- **Arbitration:** AC uses the auxiliary information sent by the SP to arbitrate disputed bills.

4.2. System Initialization

In our scheme, we assume that RKG is responsible for bootstrapping the entire system and generating system parameters. The parameters generation process operates as follows:

1. PKG generates a bilinear mapping e: $G \times G \rightarrow G_\tau$, where $G = \langle g \rangle$ and G_τ are two cyclic groups of prime order q;
2. PKG chooses a secure cryptographic hash function $H : \{0,1\}^* \rightarrow G$.

4.3. Registration

At this phase, $SM_{ij}(i = 1, 2, \cdots, k, j = 1, 2, \cdots, n)$, $FN_i(i = 1, 2, \cdots, k)$, AC and the SP make registration requests. RKG generates and distributes keys for them. Precisely, the Registration phase includes the following steps:

1. SM_{ij} registration: For each SM_{ij}, RKG selects $x_{ij} \in Z_q^*$ randomly as the private key of SM_{ij} and computes $pk_{ij} = g^{x_{ij}}$, which is set as the public key.
2. FN_i registration: For each FN_i, RKG randomly selects $x_{FN_i} \in Z_q^*$ as the private key of FN_i and computes $pk_{FN_i} = g^{x_{FN_i}}$, which is set as the public key.
3. AC registration: RKG randomly selects $x_{AC} \in Z_q^*$ as the private key of AC and computes $pk_{AC} = g^{x_{AC}}$, which is set as the public key.
4. SP registration: RKG randomly selects $x_{SP} \in Z_q^*$ as the private key of the SP and computes $pk_{SP} = g^{x_{SP}}$, which is set as the public key.

RKG securely transmits the private keys of all entities to them through the secret channel (as in the scheme of [13], we assume that the adversary cannot capture the private key transmitted in the secret channel), computes the common public key as $PK = pk_{AC}.pk_{SP} = g^{x_{AC}+x_{SP}}$ and finally publishes the system parameters par = $(G, G_\tau, e, q, g, pk_{ij}, pk_{FN_i}, pk_{AC}, pk_{SP}, PK, H)$.

4.4. Reports Generation

In our scheme, we suppose that one billing cycle consists of t report upload time slots, and in time slot m, SM_{ij} generates $\left(m_1^{ij,m}, m_2^{ij,m}, \cdots, m_l^{ij,m}\right)$ and periodically sends it to the corresponding FN_i. To prevent users' private information from being leaked, SM_{ij} encrypts data before sending it to FN_i as follows.

1. Based on the encoding function proposed in [6], SM_{ij} first constructs $\left(m_1^{ij,m}, m_2^{ij,m}, \cdots, m_l^{ij,m}\right)$ into $c_{ij,m}$ as:

$$c_w^{ij,m} = 0^{\xi} \left\| (m_w^{ij,m})_2 \right\| 0^{\lambda}, w = 1, 2, \cdots, l,$$
$$c_{ij,m} = c_1^{ij,m} + c_2^{ij,m} + \cdots + c_l^{ij,m}.$$

where

$$\xi = [\log_2(n) + z] \cdot (l - w),$$
$$\lambda = [\log_2(n) + z] \cdot (w - 1),$$

and z is the maximum number of bits of $\left(m_1^{ij,m}, m_2^{ij,m}, \cdots, m_l^{ij,m}\right)$; 0^{ξ} represents 0 with the length of ξ bits; and 0^{λ} represents 0 with the length of λ bits. The detailed construction process for $l = 4$ is given in Figure 3.

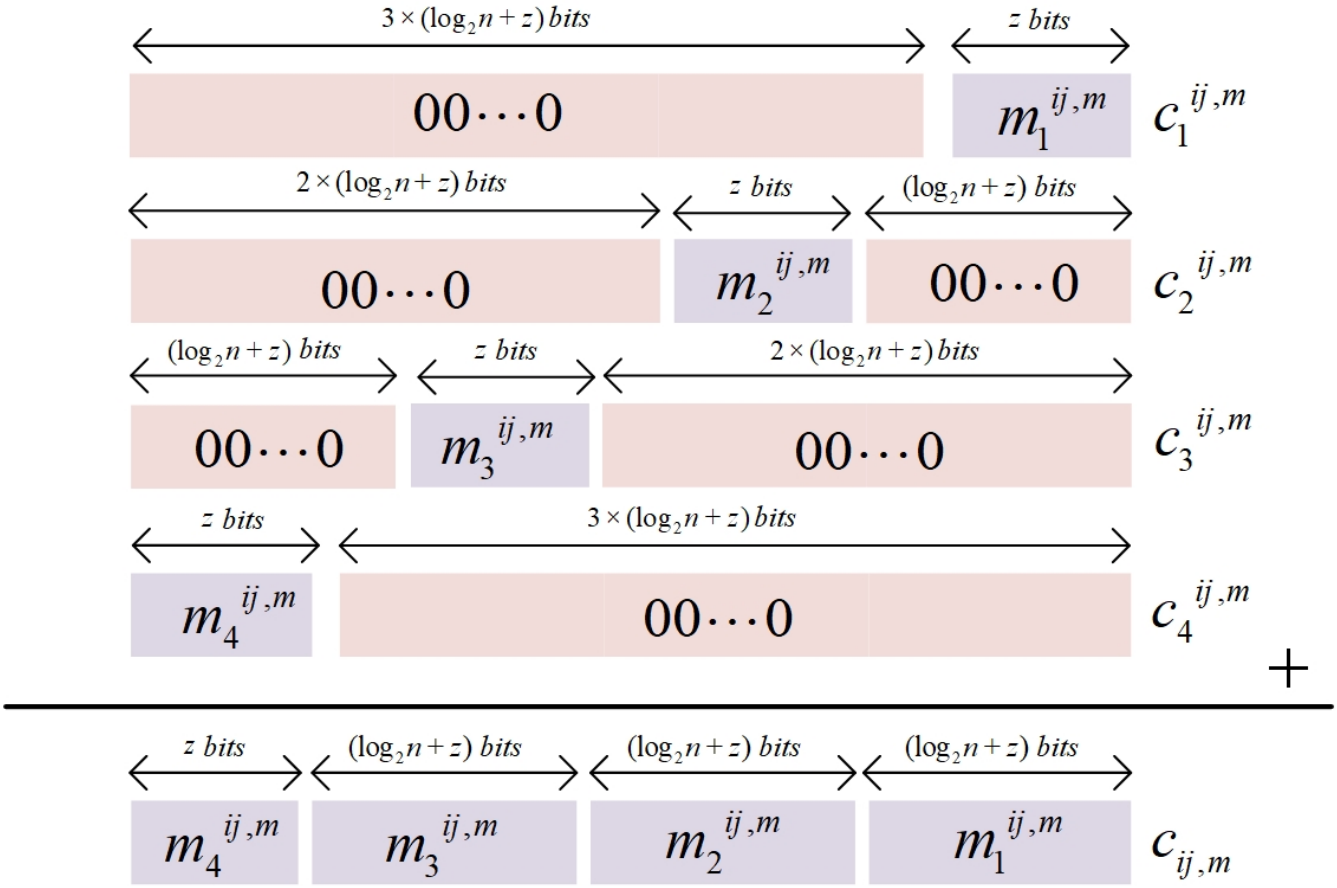


Figure 3. The detailed form of $c_{ij,m}$ ($l = 4$).

2. SM_{ij} picks a number $r_{ij,m} \in Z_q^*$ at random, and computes ciphertext $C_{ij,m} = (u_{ij,m}, v_{ij,m}) = (g^{r_{ij,m}}, PK^{r_{ij,m}} g^{c_{ij,m}})$.
3. SM_{ij} employs its private key x_{ij} to generate the signature $\sigma_{ij,m} = H(ID_{SM_{ij}} || u_{ij,m} || v_{ij,m} || T_{ij,m})^{x_{ij}}$, where $T_{ij,m}$ is a timestamp, which is used to defend against potential replay attacks.
4. SM_{ij} sends $\mathcal{T}_{ij,m} = (ID_{SM_{ij}}, u_{ij,m}, v_{ij,m}, T_{ij,m}, \sigma_{ij,m})$ to the corresponding FN_i.

4.5. Data Aggregation

1. Upon receiving $\mathcal{T}_{ij,m}$ from all covered SMs, FN_i performs the batch verification to verify all reports as

$$e(\prod_{j=1}^{n}\sigma_{ij,m}, g) = \prod_{j=1}^{n} e(H(ID_{SM_{ij}}||u_{ij,m}||v_{ij,m}||T_{ij,m}), pk_{ij}). \quad (1)$$

   If Equation (1) holds, it means that all reports are valid.
2. FN_i aggregates n ciphertexts as $C_{i,m} = (u_{i,m}, v_{i,m}) = (\prod_{j=1}^{n} u_{ij,m}, \prod_{j=1}^{n} v_{ij,m})$.
3. FN_i uses its private key x_{FN_i} to generate the signature

$$\sigma_{FN_{i,m}} = H(ID_{FN_i} \parallel u_{i,m} \parallel v_{i,m} \parallel T_{FN_{i,m}})^{x_{FN_i}}.$$

4. FN_i sends $\mathcal{T}_{i,m} = (ID_{FN_i}, u_{i,m}, v_{i,m}, T_{FN_{i,m}}, \sigma_{FN_{i,m}})$ to AC.

4.6. Data Reading

1. When it receives $\mathcal{T}_{i,m}$ from all covered FNs, AC performs the batch verification to verify whether all $\mathcal{T}_{i,m}$ satisfy the following condition:

$$e(\prod_{i=1}^{k}\sigma_{FN_{i,m}}, g) = \prod_{i=1}^{k} e(H(ID_{FN_i}||u_{i,m}||v_{i,m}||T_{FN_{i,m}}), pk_{FN_i}). \quad (2)$$

2. After checking the validity, AC uses its private key to perform pre-decryption as $C'_{i,m} = \frac{v_{i,m}}{(u_{i,m})^{x_{AC}}}$.
3. AC utilizes its private key x_{AC} to generate the signature

$$\sigma_{AC_{i,m}} = H(ID_{AC} \parallel u_{i,m} \parallel C'_{i,m} \parallel T_{AC_{i,m}})^{x_{AC}}.$$

4. AC sends $\mathcal{T}_{AC_{i,m}} = (ID_{AC}, u_{i,m}, C'_{i,m}, T_{AC_{i,m}}, \sigma_{AC_{i,m}})$ to SP.
5. Upon receiving $\mathcal{T}_{AC_{i,m}}$ from AC, the SP accepts $\mathcal{T}_{AC_{i,m}}$ if it satisfies the following condition:

$$e(\prod_{i=1}^{k}\sigma_{AC_{i,m}}, g) = \prod_{i=1}^{k} e(H(ID_{AC} \parallel u_{i,m} \parallel C'_{i,m} \parallel T_{AC_{i,m}}), pk_{AC}). \quad (3)$$

6. After verification, the SP computes

$$g^{\sum_{j=1}^{n} c_{ij,m}} = \frac{C'_{i,m}}{(u_{i,m})^{x_{SP}}}. \quad (4)$$

   After retrieving $g^{\sum_{j=1}^{n} c_{ij,m}}$, the SP first applies the pollard lambda method [35] to obtain $\sum_{j=1}^{n} c_{ij,m}$ and then utilizes the decoding function in [6] to decode $\sum_{j=1}^{n} c_{ij,m}$ as $\sum_{j=1}^{n} c_{ij,m} = \sum_{j=1}^{n} m_l^{ij,m} || \sum_{j=1}^{n} m_{l-1}^{ij,m} || \cdots || \sum_{j=1}^{n} m_2^{ij,m} || \sum_{j=1}^{n} m_1^{ij,m}$. The length of $\sum_{j=1}^{n} m_w^{ij,m} (w = 1, 2, \cdots l)$ is $(\log_2 n + z)$, which represents the total amount of data of the same type consumed by all users covered by FN_i in time slot m. The specific form of $\sum_{j=1}^{n} c_{ij,m}$ and the decoding process with $l = 4$ are shown in Figure 4.

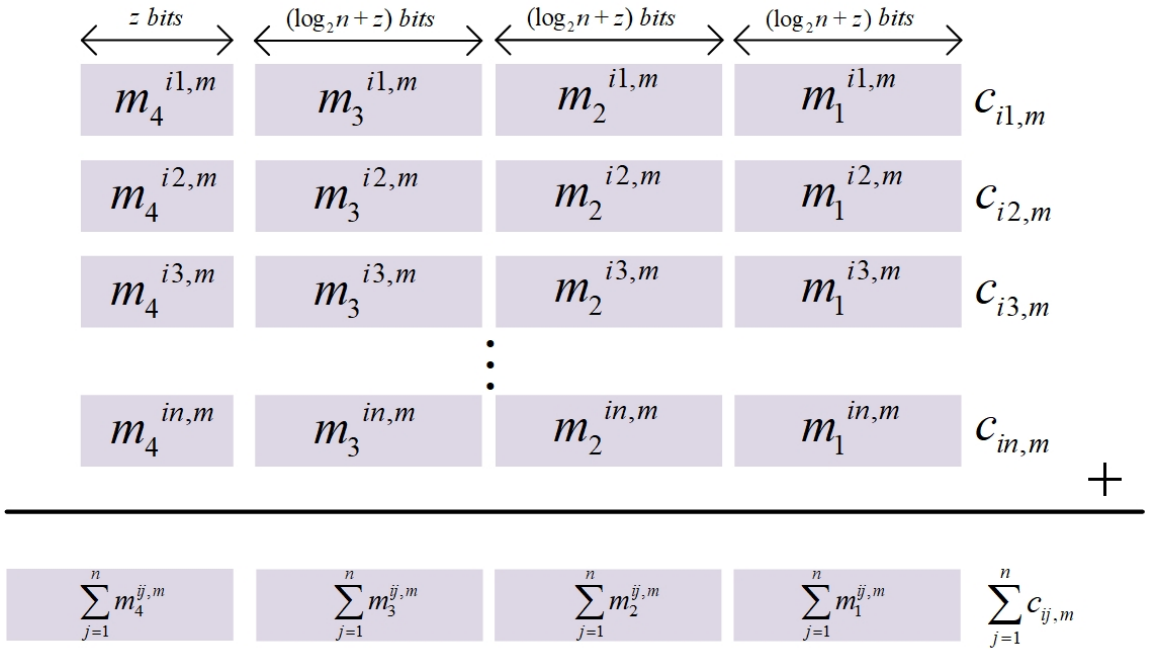


Figure 4. The detailed form of $\sum_{j=1}^{n} c_{ij,m}$ $(l = 4)$.

Obviously, the SP can obtain the total electricity consumption data of all users in each sub-region during time slot $m(m = 1, 2, \cdots, t)$. The SP analyses the obtained data to adjust generation and distribution, then anticipates the electricity price p_{m+1} for the next time slot and broadcast p_{m+1} before the start of the next time slot (giving a specific pricing process is not our focus). Users can schedule their electricity consumption according to the price in different times of the day.

4.7. Billing

The billing process consists of three parts, that is, (1) Generation and Aggregation of the Electricity Billing Ciphertext; (2) Pre-Decryption of the Electricity Billing Ciphertext; and (3) Bills Generation.

4.7.1. Generation and Aggregation of the Electricity Billing Ciphertext

1. Based on $(u_{ij,m}, v_{ij,m})$ and the price p_m, FN_i can calculate the electricity billing ciphertext $(C_{ij.m})^{p_m} = (U_{ij,m}, V_{ij,m}) = ((u_{ij,m})^{p_m}, (v_{ij,m})^{p_m})$. Essentially, $(C_{ij,m})^{p_m}$ is the encrypted form of $c_{ij,m} \cdot p_m$.
2. At the end of a billing cycle, FN_i aggregates the electricity billing ciphertext of users at all time slots to obtain the aggregated ciphertext as follows: $B_{ij} = (U_{ij}, V_{ij}) = (\prod_{m=1}^{t} U_{ij.m}, \prod_{m=1}^{t} V_{ij.m})$.
3. FN_i employs its private key to generate the signature

$$\sigma_{FN'_{ij}} = H(ID_{FN_i} \parallel ID_{SM_{ij}} || U_{ij} \parallel V_{ij} \parallel T_{FN'_i})^{x_{FN_i}}.$$

4. FN_i sends $\mathcal{T}'_{ij} = (ID_{FN_i}, ID_{SM_{ij}}, U_{ij}, V_{ij}, T_{FN'_i}, \sigma_{FN'_i})$ to AC.

4.7.2. Pre-Decryption of the Electricity Billing Ciphertext

1. Upon receiving the $\mathcal{T}'_{ij}$ from FN_i, AC performs the batch verification to verify all $\mathcal{T}'_{ij}$ according to the following equation:

$$e(\prod_{j=1}^{n}\sigma_{FN'_{ij}}, g) = \prod_{j=1}^{n} e(H(ID_{FN_i} \| ID_{SM_{ij}} \| U_{ij} \| V_{ij} \| T_{FN_{i'}}), pk_{FN_i}). \quad (5)$$

2. If Equation (5) holds, AC uses its private key to perform pre-decryption as follows: $B'_{ij} = \frac{V_{ij}}{(U_{ij})^{x_{AC}}}$.
3. AC utilizes its private key to generate the signature

$$\sigma_{AC'_{ij}} = H(ID_{AC} \parallel ID_{SM_{ij}} || U_{ij} \parallel B'_{ij} \parallel T_{AC'})^{x_{AC}}.$$

4. AC sends $\mathcal{T}'_{AC} = (ID_{AC}, ID_{SM_{ij}}, U_{ij}, B'_{ij}, T_{AC'}, \sigma_{AC'_i})$ to the SP.

4.7.3. Bills Generation

1. Upon receiving packet $\mathcal{T}'_{AC}$ from AC, the SP checks if the following equation holds:

$$e(\prod_{j=1}^{n}\sigma_{AC'_{ij}}, g) = \prod_{j=1}^{n} e(H(ID_{AC} \parallel ID_{SM_{ij}} \parallel U_{ij} \parallel B_{ij'} \parallel T_{AC'}), pk_{AC}). \quad (6)$$

2. If Equation (6) is satisfied, the SP uses its private key to calculate $g^{\sum_{m=1}^{t} c_{ij.m} \cdot p_m}$ as follows:

$$g^{\sum_{m=1}^{t} c_{ij.m} \cdot p_m} = \frac{B'_{ij}}{(U_{ij})^{x_{SP}}}. \quad (7)$$

After obtaining $g^{\sum_{m=1}^{t} c_{ij.m} \cdot p_m}$, the SP can obtain $\sum_{m=1}^{t} c_{ij.m} \cdot p_m$ using the Pollard lambda method [35]. Then, using the decoding function in [6], the SP can obtain the user's bill *Bill* for a billing cycle and charge the user according to *Bill*.

4.8. Arbitration

When users and the SP have disputes over electricity bills, AC is responsible for arbitrating the disputes, which consists of the following five steps:

1. Upon receiving the arbitration request, for the disputed electricity billing ciphertext, AC can pre-decrypt it with the private key using $B'_{ij} = \frac{V_{ij}}{(U_{ij})^{x_{AC}}}$.
2. The SP uses its private key to compute $W = (U_{ij})^{x_{SP}}$ and generate the signature $\sigma_{SP} = H(ID_{SP}||W||Bill||T_{SP})^{x_{SP}}$, respectively.
3. The SP sends $\mathcal{T}_{SP} = (ID_{SP}, W, Bill, T_{SP}, \sigma_{SP})$ to AC.
4. Upon receiving $\mathcal{T}_{SP}$ from the SP, AC verifies $\mathcal{T}_{SP}$ with the following equation:

$$e(\sigma_{SP}, g) = e(H(ID_{SP}||W||Bill||T_{SP}), pk_{SP}). \quad (8)$$

5. If Equation (8) is satisfied, AC computes

$$g^{\sum_{m=1}^{t} r_{ij.m} \cdot p_m} = \frac{B'_{ij}}{W}. \quad (9)$$

After obtaining $g^{\sum_{j=1}^{t} r_{ij,m} \cdot p_m}$, $\sum_{m=1}^{t} r_{ij,m} \cdot p_m$ is obtained by using the Pollard lambda method [35]. Then, AC can arbitrate effectively through comparing with *Bill* sent by the user and the SP.

Furthermore, if users want to query the electricity billing for a certain period of time (for example, from 10 June to 20 June), they can initiate query requests to the SP. The SP can obtain the aggregated ciphertexts for the time period through the FN, then decrypt them with the help of AC and finally send bills to users. Note that this can only be initiated by the user, and the SP cannot query a user's electricity bill in a particular period.

5. Soundness and Security Analysis

In this section, we illustrate that PPDB is sound and meets the security requirements.

Theorem 1. *The proposed PPDB construction is sound.*

Proof of Theorem 1. To prove the soundness of PPDB, we just need to prove that Equations (1)–(9) hold.

1. If all reports $T_{ij,m} = (ID_{SM_{ij}}, u_{ij,m}, v_{ij,m}, T_{ij,m}, \sigma_{ij,m})(i = 1, 2, \cdots k, j = 1, 2, \cdots n)$ sent by the SMs to the corresponding $FN_i(i = 1, 2, \cdots k)$ are valid, then we see the Equation (1) is established as follows

$$\begin{aligned} e(\prod_{j=1}^{n} \sigma_{ij,m}, g) &= e(\prod_{j=1}^{n} H(ID_{SM_{ij}} || u_{ij,m} || v_{ij,m} || T_{ij,m})^{x_{ij}}, g) \\ &= \prod_{j=1}^{n} e(H(ID_{SM_{ij}} || u_{ij,m} || v_{ij,m} || T_{ij,m})^{x_{ij}}, g) \\ &= \prod_{j=1}^{n} e(H(ID_{SM_{ij}} || u_{ij,m} || v_{ij,m} || T_{ij,m}), pk_{ij}). \end{aligned}$$

2. If all reports $\mathcal{T}_{i,m} = (ID_{FN_i}, u_{i,m}, v_{i,m}, T_{FN_{i,m}}, \sigma_{FN_{i,m}})(i = 1, 2, \cdots k)$ sent by $FN_i(i = 1, 2, \cdots k)$ to AC are valid. Obviously, the Equation (2) holds, as follows:

$$\begin{aligned} e(\prod_{i=1}^{k} \sigma_{FN_{i,m}}, g) &= e(\prod_{i=1}^{k} H(ID_{FN_i} \| u_{i,m} \| v_{i,m} \Big\| T_{FN_{i,m}})^{x_{FN_i}}, g) \\ &= \prod_{i=1}^{k} e(H(ID_{FN_i} \| u_{i,m} \| v_{i,m} \Big\| T_{FN_{i,m}})^{x_{FN_i}}, g) \\ &= \prod_{i=1}^{k} e(H(ID_{FN_i} \| u_{i,m} \| v_{i,m} \Big\| T_{FN_{i,m}}), pk_{FN_i}). \end{aligned}$$

3. If all reports $\mathcal{T}_{AC_{i,m}} = (ID_{AC}, u_{i,m}, C'_{i,m}, T_{AC_{i,m}}, \sigma_{AC_{i,m}})(i = 1, 2, \cdots k)$ sent by AC to the SP are valid, then the left of Equation (3) can be expanded as

$$\begin{aligned} e(\prod_{i=1}^{k} \sigma_{AC_{i,m}}, g) &= e(\prod_{i=1}^{k} H(ID_{AC} \parallel u_{i,m} \parallel C'_{i,m} \parallel T_{AC_{i,m}})^{x_{AC}}, g) \\ &= \prod_{i=1}^{k} e(H(ID_{AC} \parallel u_{i,m} \parallel C'_{i,m} \parallel T_{AC_{i,m}})^{x_{AC}}, g) \\ &= \prod_{i=1}^{k} e(H(ID_{AC} \parallel u_{i,m} \parallel C'_{i,m} \parallel T_{AC_{i,m}}), pk_{AC}). \end{aligned}$$

That means that Equation (3) holds.

4. We demonstrate the soundness of Equation (4) by expanding the left side of Equation (4) in detail:

$$
\begin{aligned}
g^{\sum_{j=1}^{n} c_{ij,m}} &= \frac{(pk_{SP})^{\sum_{j=1}^{n} r_{ij,m}} g^{\sum_{j=1}^{n} c_{ij,m}}}{(pk_{SP})^{\sum_{j=1}^{n} r_{ij,m}}} \\
&= \frac{(pk_{SP})^{\sum_{j=1}^{n} r_{ij,m}} g^{\sum_{j=1}^{n} c_{ij,m}}}{(g^{x_{SP}})^{\sum_{j=1}^{n} r_{ij,m}}} \\
&= \frac{(pk_{SP})^{\sum_{j=1}^{n} r_{ij,m}} g^{\sum_{j=1}^{n} c_{ij,m}}}{(g^{\sum_{j=1}^{n} r_{ij,m}})^{x_{SP}}} \\
&= \frac{C'_{i,m}}{(\prod_{j=1}^{n} u_{ij,m})^{x_{SP}}} \\
&= \frac{C'_{i,m}}{(u_{i,m})^{x_{SP}}}.
\end{aligned}
$$

With the above expansion, we can see that Equation (4) holds.

Since the Equations (5)–(9) are constructed similarly to the previous equations, their soundness can be proven in the same way, and we do not prove them one by one. Through the above analysis, it is clear that the proposed PPDB is sound. □

Theorem 2. *Supposing that the DDH assumption holds, the proposed PPDB is IND-CPA secure under it. In other words, the proposed PPDB can provide strong confidentiality for users' electricity consumption data against the chosen plaintext attack.*

Proof of Theorem 2. In PPDB, in order to guarantee the confidentiality of users' privacy, SM_{ij} first utilizes the coding function [6] to process the l-dimensional electricity consumption data $\left(m_1^{ij,m}, m_2^{ij,m}, \cdots, m_l^{ij,m}\right)$ into one-dimensional data $c_{ij,m}$ and then encrypt $c_{ij,m}$ as $C_{ij,m} = (u_{ij,m}, v_{ij,m}) = (g^{r_{ij,m}}, PK^{r_{ij}} g^{c_{ij,m}})$. In fact, $C_{ij,m}$ is a valid ciphertext of hte ElGamal cryptosystem. Since the ElGamal cryptosystem is semantic secure under the DDH assumption [36], the data $\left(m_1^{ij,m}, m_2^{ij,m}, \cdots, m_l^{ij,m}\right)$ are also semantic secure and confidential. As a result, it is impossible for any entity that has no knowledge of private key x_{ij} to reveal any information concerning the consumption data of the corresponding user. Even if the adversary $\mathcal{A}$ taps $C_{ij,m}$, $\mathcal{A}$ still can not reveal the corresponding plaintext information.

Upon receiving all the reports sent by all covered SMs, FN_i cannot recover each report; instead, FN_i just computes $C_{i,m} = (\prod_{j=1}^{n} u_{ij,m}, \prod_{j=1}^{n} v_{ij,m}) = (g^{\sum_{j=1}^{n} r_{ij,m}}, (PK)^{\sum_{j=1}^{n} r_{ij,m}} g^{\sum_{j=1}^{n} c_{ij,m}})$ to perform the aggregation of the ciphertext data. As a result, even if $\mathcal{A}$ invades into FN_i's database, $\mathcal{A}$ cannot obtain the user's usage data $\left(m_1^{ij,m}, m_2^{ij,m}, \cdots, m_l^{ij,m}\right)$. Similarly, the reports generated by AC are still valid ciphertexts in the ElGamal cryptosystem, so even if $\mathcal{A}$ hacks into AC's database, $\mathcal{A}$ cannot obtain personal usage data $\left(m_1^{ij,m}, m_2^{ij,m}, \cdots, m_l^{ij,m}\right)$. Finally, after receiving the report from AC, the SP can only decrypt it to obtain the total amount of power usage of the same type in one area $\sum_{j=1}^{n} m_w^{ij,m} (w = 1, 2, \cdots, l)$; the SP still cannot infer the consumption data of each user.

The above discussion shows that neither external nor internal adversaries can access the users' usage data, and therefore, the proposed PPDB can provide strong confidentiality. □

Theorem 3. *Supposing the CDH assumption holds, the proposed PPDB is EU-CMA secure under it. In other words, the proposed PPDB can guarantee the integrity and authentication of users' electricity consumption data against the chosen message attack.*

Proof of Theorem 3. In the threat model, we suppose that $\mathcal{A}$ can launch a number of active attacks (for example, tampering, forgery, replay) to compromise the source authentication and data integrity, thus threatening the stability of the smart grid.

In order to ensure that the proposed PPDB has strong security against active attacks launched by adversaries, we adopt timestamps to protect against replay attacks. For the report $\mathcal{T}_{ij,m} = (ID_{SM_{ij}}, u_{ij,m}, v_{ij,m}, T_{ij,m}, \sigma_{ij,m}, pk_{ij})$ sent by SM_{ij}, the corresponding FN_i first checks the timestamp and identity, and then, verifying the integrity of the report by checking whether Equation (1) holds, from the construction of the equation, it can be seen if any element of the report is tampered with, which will cause the equation to fail. Thus, the integrity of the report can be guaranteed. In general, the integrity of reports sent by entities can be verified by the corresponding entities through checking whether the corresponding equations are satisfied.

We employ the BLS signature [37] to sign the reports sent by each entity. Since the BLS signature was proven to be secure under the CDH assumption [37], the adversary cannot forge a valid ciphertext without knowing the entity's private key. Our scheme authenticates the source of the reports, thus ensuring that the reports received are from legitimate entities. □

From the above analysis, it is clear that the PPDB scheme can provide integrity and authentication for users' electricity consumption data.

6. Comparison and Evaluation

6.1. Features Comparison

In this section, we compare our PPDB with the schemes of [23,27] in terms of their features; more details are shown in the Table 2.

Table 2. Features comparison.

	Multi-Dimensional Data	Confidentiality	Integrity	Authentication	Billing	Arbitration
Li et al.'s scheme [23]	✗	✓	✓	✓	✓	✗
Zuo et al.'s scheme [27]	✓	✓	✓	✓	✗	✗
Our scheme	✓	✓	✓	✓	✓	✓

In [23], Li et al. utilized the Paillier cryptosystem and Horner's rule to construct a privacy-preserving dynamic billing scheme, called PPCSB. In their scheme, each SM generates only one type of data; thus, CC can only obtain the total electricity bill of the user. Compared to scheme [23], our PPDB aggregates multi-dimensional data to obtain the electricity bill generated by each type of electrical equipment.

In PPDB and [27], each SM generates l types of data for more accurate fine-grained analysis. To reduce the complexity, we use a binary encoding function to construct multi-dimensional data, which can effectively save ciphertext space compared with the [27]. Meanwhile, Zuo et al. did not consider dynamic billing based on real-time electricity prices.

All three schemes meet the security requirements for confidentiality, integrity and authentication. It is worth noting that neither scheme [23] nor scheme [27] has designed an arbitration mechanism. By the above comparison, our PPDB is more diverse in features and is more suitable for practical deployment.

6.2. Computation Overhead

We first take a sub-region (with n SMs) as an example to analyse and compare the computational overhead at each side of the schemes of [23,27] and PPDB theoretically. For the sake of fairness, our comparison focuses on the Reports Generation, Data Aggregation and Data Reading phases, which are present in all three schemes. Our analysis concentrates only on the most time-consuming operations, that is, encryption under the Paillier cryptosystem (denoted as $T_{C-Paillier}$), decryption under the Paillier cryptosystem (denoted as $T_{D-Paillier}$), encryption under the ElGamal cryptosystem (denoted as $T_{C-ElGamal}$), decryption under the ElGamal cryptosystem (denoted as $T_{D-ElGamal}$), bilinear pairing (denoted as T_B) and exponentiation operation (denoted as T_E).

In our PPDB, each SM requires $T_{C-ElGamal}$ to generate the ciphertext and another T_E for signature generation. After receiving the reports sent by n SMs, the FN first operates a batch verification algorithm to check the validity of all reports, which contains $(n+1)T_B$. In addition, the FN requires T_E to generate a signature. For AC, it first performs $2T_B$ to verify the validity of the report sent by FN, $T_{C-ElGamal}$ to pre-decrypt it and T_{E_1} to generate the signature. When it receives the report from AC, the SP first takes $2T_B$ to verify it and then $T_{D-ElGamal}$ to decrypt it. In summary, the total computation overheads at the SM side, FN side, AC side and SP side is $T_{C-ElGamal}+T_E$, $(n+1)T_B+T_E$, $2T_B+T_{C-ElGamal}+T_E$ and $2T_B+T_{C-ElGamal}$, respectively.

In [23], Li et al. only presented the dynamic billing process. For theoretical comparison, we supplement the electricity consumption statistics process for [23]. According to our supplement, in [23], each SM requires $T_{C-Paillier}$ to generate ciphertext and T_E to generate signature. Hence, the total computational overhead at each SM side is $T_{C-Paillier}+T_E$. The FN takes $(n+1)T_B+T_E$ to batch validate all reports received from the SMs and generates signatures. When receiving the report sent by the FN, AC performs $2T_B$ to verify the validity of the report and $T_{D-Paillier}$ to decrypt it. As a result, the total computation overhead at the AC side is $2T_B+T_{D-Paillier}$.

In Zuo et al.'s scheme [27], each SM requires $T_{C-ElGamal}$ to generate the ciphertext and T_E to generate signatures. In addition, in the Data Reading phase, each SM needs to perform distributed decryption, which consists of $2T_E$. In total, the computation overhead at each SM side is $T_{C-ElGamal}+3T_E$. Same as scheme [23], in [27], the computation overhead at the FN side is $(n+1)T_B+T_E$. In the Data Reading, the SP needs $2(n+1)T_B$ to verify signatures and $T_{C-ElGamal}$ to decrypt the aggregated ciphertext. So, the total computation overhead at the SP side is $2(n+1)T_B+T_{C-ElGamal}$.

Theoretical comparison of computation overhead between our proposed PPDB and schemes [23,27] is shown in Table 3. We can see that our scheme has low computation overhead at the SP side compared to the scheme [27].

Table 3. Theoretical comparison of computational overhead.

	SM Side	FN Side	AC Side	SP Side
Li et al.'s scheme [23]	$T_{C-Paillier}+T_E$	$(n+1)T_B+T_E$	$2T_B+T_{D-Paillier}$	/
Zuo et al.'s scheme [27]	$T_{C-ElGamal}+3T_E$	$(n+1)T_B+T_E$	/	$2(n+1)T_B+T_{C-ElGamal}$
Our scheme	$T_{C-ElGamal}+T_E$	$(n+1)T_B+T_E$	$2T_B+T_{C-ElGamal}+T_E$	$2T_B+T_{C-ElGamal}$

$T_{C-Paillier}$: encryption under the Paillier cryptosystem; $T_{D-Paillier}$: decryption under the Paillier cryptosystem; $T_{C-ElGamal}$: encryption under the ElGamal cryptosystem; $T_{D-ElGamal}$: decryption under the ElGamal cryptosystem; T_B: bilinear pairing; T_E: exponentiation operation.

We proceed to compare the computational overhead of the schemes [23,27] and PPDB by conducting extensive experiments using the Java Pairing-Based Cryptography Library on a computer with a Microsoft Windows 11 operating system, an Intel Core i3-12100 CPU @ 3.30 GHz processor and 16 GB memory.

Figure 5 presents the performance of our PPDB in the Registration, Reports Generation, Data Aggregation, Data Reading, Billing and Arbitration phases, where the number of SMs

and time slots are set to $n = 10, t = 10$, respectively. In the Registration phase, RKG runs the Key generation algorithm to generate keys for entities registered in the system. When $n = 10$, this process takes about 110 ms. In the Reports Generation phase, each SM first runs Encryption algorithm to encrypt data and then runs Signature algorithm to generate a signature. This process contains $(T_{C-ElGamal} + T_E)$ and takes about 440 ms. After receiving reports from all SMs in the coverage area, FN runs the Batch verification algorithm to check the validity of these reports, and then runs the Signature algorithm to generate a signature after aggregating the valid ciphertexts. At this phase, PPDB has the same computational complexity (i.e., requires $(n+1)T_B + T_E$) as in scheme [23,27], which takes about 450 ms. In the Data Reading phase, AC first runs the pre-decryption algorithm to decrypt the report sent by the FN, and then the SP runs the decryption algorithm to obtain the final result. This process consists of $(4T_B + 2T_{C-ElGamal} + T_E)$ and takes about 60 ms. Our scheme also considers real-time-price-based billing and arbitration of disputed bills. When $n = 10$, $t = 10$, the Billing phase takes about 840 ms and the Arbitration phase about 65 ms.

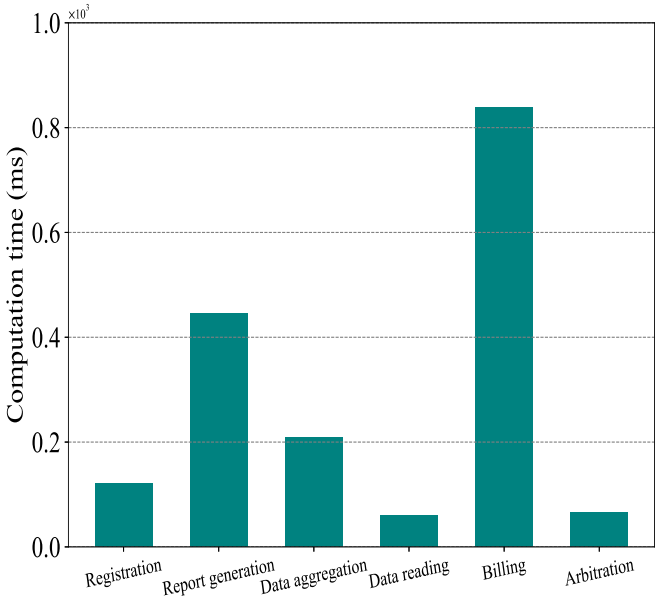


Figure 5. Computation time in each phase (n = 10).

Upon receiving reports from all SMs in the coverage area, the FN can verify the reports by two different verification methods, which are single verification and batch verification, and our PPDB employs batch validation to verify all reports. The efficiency comparison result of the two different verification methods is shown in Figure 6, where the number of SMs is set to $n = 10, 20, 30, \cdots, 100$. It can be seen that when the number of SMs is the same, the computation time required for batch verification is smaller than that required for single verification, and with the increase in n, the time difference between batch verification and single verification will increase. When $n = 100$, single validation takes about 0.6×10^3 ms more than that for batch verification. The reasons for this are: when the number of SMs is n, in addition to the less time-consuming multiplication and hash operations, single verification needs to perform $2nT_B$, while the batch verification only needs $(n+1)T_B$. When n is large ($n \geq 60$), the computation time required for multiplication and hash operations can be ignored; in this case, the computation time required for batch verification and single verification gradually shows a linear trend.

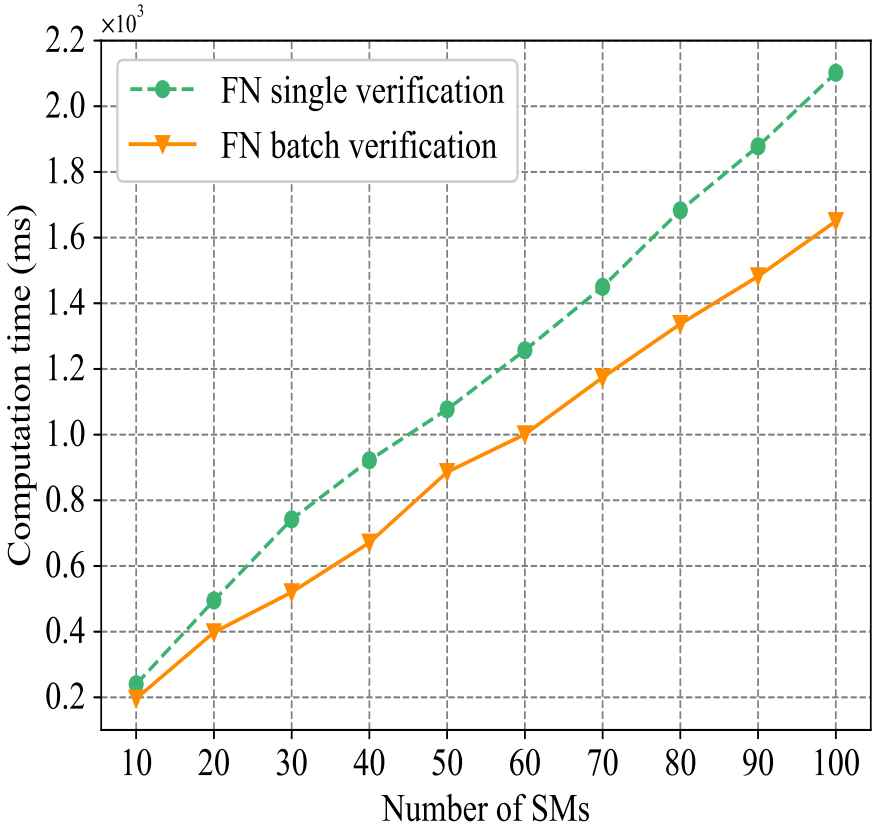
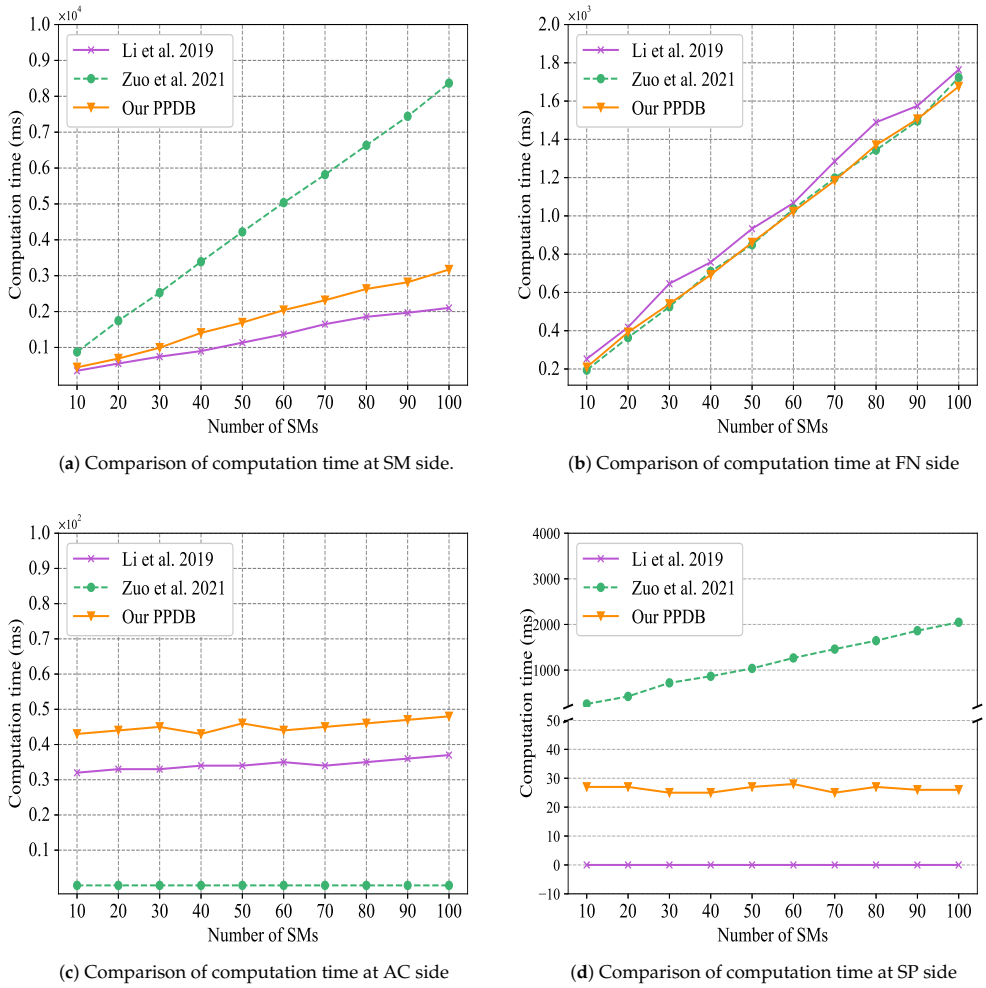


Figure 6. Comparison of computation time for batch verification and single verification by FN.

To further evaluate the performance of PPDB, in our simulations, we compared the computation time of [23,27] and PPDB at each side in various cases (i.e., the number of SMs is set to $n = 10, 20, 30, \cdots, 100$). The experimental results are shown in Figure 7, where Figure 7a–d show the comparison of computation time of three schemes at the SM side, FN side, AC side and SP side, respectively. Although the computation time of PPDB is higher than that of scheme [23] at the SM side, in our scheme, each SM processes multi-dimensional data, while those in scheme [23] encrypts only one-dimensional data. From Figure 7b, it can be seen that the three schemes have almost the same computation time at the FN side because they have the same computational complexity at the FN side. According to Figure 7c,d, we can see that the computation time of PPDB is only a little higher than that of scheme [23] at the AC side. However, the computation time of PPDB at the SP side is far less than that of [27]. The reason is: in PPDB, when the number of SMs is n, SP needs to execute $2T_B + T_{C-ElGamal}$, while in [27], SP needs to execute $2(n+1)T_B + T_{C-ElGamal}$. When $n = 100$, the computation time of [27] at the SP side is about 70 times that of PPDB. From the above analysis, it is easy to see that PPDB has good performance.

We also compared the computation time of scheme [23] with our scheme in the Billing phase. In our simulations, we set the number of SMs and time slots to $n = 10, 20, 30, \cdots, 100$, $t = 10, 20, 30, \cdots, 100$, respectively. Figure 8 shows the comparison results. It is not difficult to find that the computation time required for scheme [23] is much higher than that of PPDB. When $n = 100$ and $t = 100$, PPDB is about 40 times more efficient than [23].



(**a**) Comparison of computation time at SM side.

(**b**) Comparison of computation time at FN side

(**c**) Comparison of computation time at AC side

(**d**) Comparison of computation time at SP side

Figure 7. Comparison of computation time of Li et al. [23], Zuo et al. [27] and our PPDB at each side.

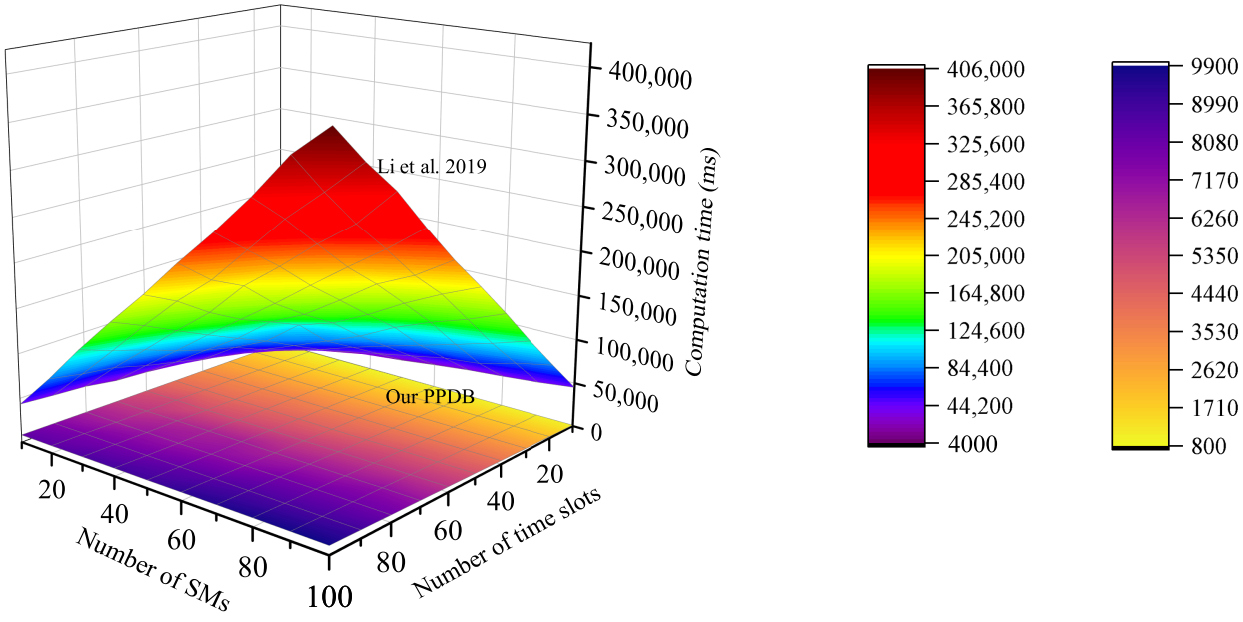


Figure 8. Comparison of computation time between Li et al. [23] and our PPDB in Billing phase.

6.3. Communication Overhead

In this section, we also take a sub-region (with n SMs) as an example to analyse and compare the communication overhead of the scheme [23,27] and PPDB. For the sake of fairness, we also focus only on the Reports Generation, Data Aggregation and Data Reading phases, phases, which are present in all three schemes. According to the construction of the schemes, we divide the communication process into five parts: (1) SM-to-FN communication; (2) FN-to-AC communication; (3) FN-to-SP communication; (4) AC-to-SP communication; and (5) SM-to-SP communication. We select $|N| = 1024$ bits for the Paillier cryptosystem and the length of the elements in G is 512 bits. In addition, we assume that the sum of the length of the identity and timestamp is 64 bits. The communication overhead of each part is given in Table 4.

Table 4. Comparison of communication overhead.

	SM-to-FN	**FN-to-AC**	**FN-to-SP**	**AC-to-SP**	**SM-to-SP**
Li et al.'s scheme [23]	2624 bits	2624 bits	/	/	/
Zuo et al.'s scheme [27]	1600 bits	/	1600 bits	/	1600 bits
Our scheme	1600 bits	1600 bits	/	1600 bits	/

In our proposed PPDB, each SM sends the report $\mathcal{T}_{ij,m} = (ID_{SM_{ij}}, u_{ij,m}, v_{ij,m}, T_{ij,m}, \sigma_{ij,m})$ to the corresponding FN, and its length is $L_{SM} = 64 + 512 + 512 + 512 = 1600$ bits. Next, we analyse the communication overhead from FN to AC. After generating a signature, FN sends the report $\mathcal{T}_{ij,m} = (ID_{SM_{ij}}, u_{ij,m}, v_{ij,m}, T_{ij,m}, \sigma_{ij,m})$ to AC. Therefore, the communication overhead from FN to AC should be $L_{FN} = 64 + 512 + 512 + 512 = 1600$ bits. In the Data Reading phase, AC sends $\mathcal{T}_{AC_{i,m}} = (ID_{AC}, u_{i,m}, C'_{i,m}, T_{AC_{i,m}}, \sigma_{AC_{i,m}})$ to the SP, the size of which is $L_{AC} = 64 + 512 + 512 + 512 = 1600$ bits. The FN collects reports for n SMs in the coverage area, AC receives one report from the FN and the SP also receives only one report from AC. Therefore, the total communication overhead of the proposed PPDB in the Reports Generation, Data Aggregation and Data Reading phases is $(nL_{SM} + L_{FN} + L_{AC})$ bits.

In the scheme [23], each SM sends $\{l_i, ID_{SMi}, TS, \sigma_i\}$ to the FN, where $l_i \in Z_{N^2}$. Thus, the communication overhead from the SM to the FN is $L_{SM'} = 64 + 2048 + 512 = 2624$ bits. The report sent from the FN to AC is in the form of $\{C, ID_{GW}, TS, \sigma\}$, and its length is $L_{FN'} = 64 + 2048 + 512 = 2624$ bits. Therefore, the total communication overhead of the scheme [23] is $(nL_{SM'} + L_{FN'})$ bits.

In the scheme [27], each SM sends $\left\{ID_i, C_i^a, C_i^a, T_i, \sigma_i\right\}$ to the FN with the size of $L_{SM''} = 64 + 512 + 512 + 512 = 1600$ bits. In the Data Reading phase, the SM also needs to send a report to the SP in the form of $\left\{ID_i, D_i, T_i^d, \sigma_i^d\right\}$, and its length is $L_{SM'''} = 64 + 512 + 512 = 1088$ bits. After generating a signature, the FN sends the report $\left\{ID_{GW}, C^a, C^b, T_{GW}, \sigma_{GW}\right\}$ to the SP. Therefore, the communication overhead from the FN to the SP should be $L_{FN''} = 64 + 512 + 512 + 512 = 1600$ bits. Therefore, the total communication overhead of the scheme [27] is $(nL_{SM''} + nL_{SM'''} + L_{FN''})$ bits.

Comparison of total communication overhead between our proposed PPDB and schemes [23,27] is shown in Figure 9. It can be observed that compared with schemes [23,27], our PPDB has the lowest communication overhead. When $n = 100$, the communication overhead of scheme [23] and scheme [27] is about 1.8 times that of PPDB.

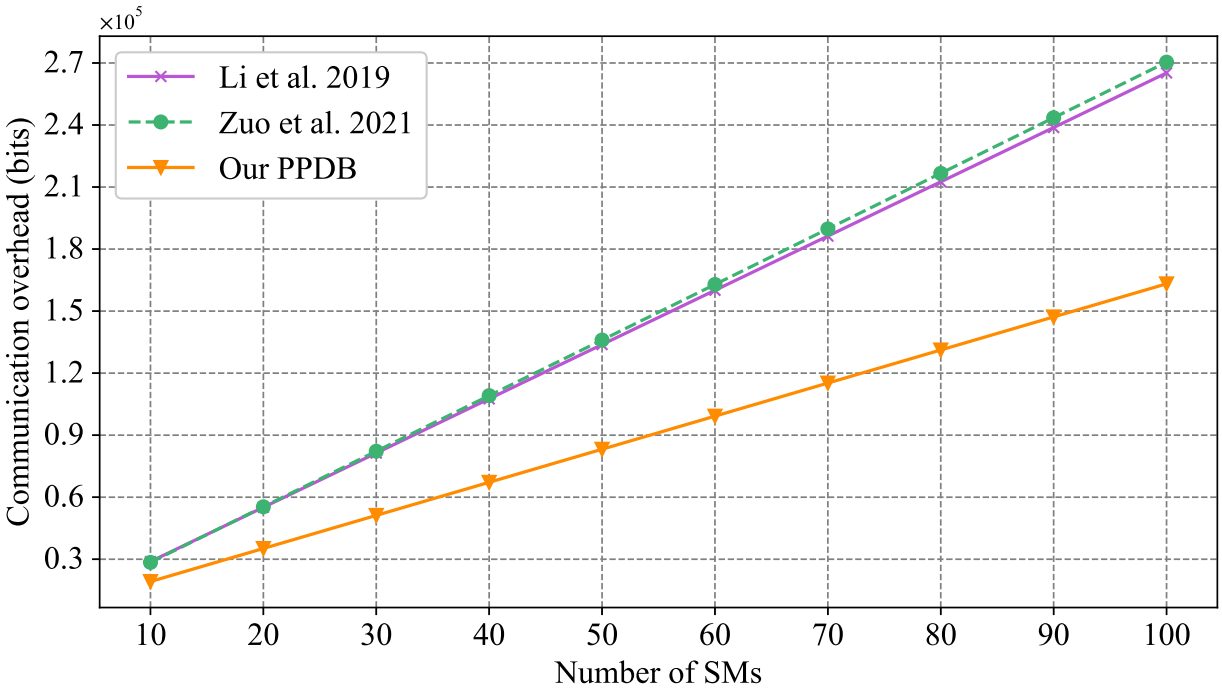


Figure 9. Comparison of communication overhead of Li et al. [23], Zuo et al. [27] and our PPDB.

7. Conclusions

This paper presents a privacy-preserving aggregation scheme in the context of FCSG. Compared with most existing smart grid schemes, our scheme allows the SP to charge users based on real-time electricity prices, and we introduce a trusted third party which can effectively arbitrate disputes between the SP and users over costs. We analyze our scheme in terms of confidentiality, authentication and integrity. The analysis shows that our scheme satisfies the security requirements. Compared with relevant schemes, PPDB greatly reduces the computation and communication overheads. Specifically, the computational efficiency of PPDB in the Billing phase has been improved by at least 40 times, and the communication overhead has been reduced by at least 38%. Considering that most of the existing privacy protection schemes assume that there is a trusted authority and a secure channel to generate and distribute keys, such assumption is vulnerable to attacks in the real world. Morover, since the trusted authority knows the key of each participant, it is easy to pose a threat to the privacy of users. Therefore, in the future work, we aim to design a scheme that does not rely on the trusted authority and secure channel, so that the constructed scheme has better robustness.

Author Contributions: Conceptualization, H.W. and Y.G.; methodology, H.W., S.T. and Y.W.; soft ware, H.W., Y.G. and Y.W.; validation, H.W., Y.G. and S.T.; formal analysis, H.W., Y.G. and S.T.; investigation, Y.G.; resources, Y.D. and H.W.; data curation, Y.D.; writing—original draft preparation, Y.G.; writing—review and editing, S.T., Y.D. and H.W.; supervision, Y.D.; project administration and funding acquisition, Y.D. All authors have read and agreed to the published version of the manuscript.

Funding: This article is supported in part by the National Key R&D Program of China under Project (2020YFB1006003), the Guangxi Natural Science Foundation under grant (2019GXNSFGA245004), the Major Key Project of PCL under grants (PCL2021A09, PCL2021A02, PCL2021A03) and the National Natural Science Foundation of China under Project (61962012,62172119, 62162017).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: We would like to thank the anonymous reviewers for their comments and suggestions that helped us improve the paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Kumari, A.; Gupta, R.; Tanwar, S. Amalgamation of blockchain and IoT for smart cities underlying 6G communication: A comprehensive review. *Comput. Commun.* **2021**, *172*, 102–118. [CrossRef]
2. Li, Y.; Wang, T.; Wang, S. Cost-efficient approximation algorithm for aggregation points planning in smart grid communications. *Wirel. Net.* **2020**, *26*, 521–530. [CrossRef]
3. Deng, L.; Gao, R. Certificateless two-party authenticated key agreement scheme for smart grid. *Inf. Sci.* **2021**, *543*, 143–156. [CrossRef]
4. Lu, R.; Liang, X.; Li, X.; Lin, X.; Shen, X. EPPA: An Efficient and Privacy-Preserving Aggregation Scheme for Secure Smart Grid Communications. *IEEE Trans. Parallel Distrib. Syst.* **2012**, *23*, 1621–1631. [CrossRef]
5. Lu, R.; Heung, K.; Lashkari, A.H.; Ghorbani, A.A. A Lightweight Privacy-Preserving Data Aggregation Scheme for Fog Computing-Enhanced IoT. *IEEE Access* **2017**, *5*, 3302–3312. [CrossRef]
6. Merad-Boudia, O.R.; Senouci, S.M. An Efficient and Secure Multidimensional Data Aggregation for Fog-Computing-Based Smart Grid. *IEEE Internet Things J.* **2021**, *8*, 6143–6153. [CrossRef]
7. Zhan, Y.; Zhou, L.; Wang, B.; Duan, P.; Zhang, B. Efficient Function Queryable and Privacy Preserving Data Aggregation Scheme in Smart Grid. *IEEE Trans. Parallel Distrib. Syst.* **2022**, *33*, 3430–3441. [CrossRef]
8. Li, J.; Gu, C.; Xiang, Y.; Li, F. Edge-cloud Computing Systems for Smart Grid: State-of-the-art, Architecture, and Applications. *J. Mod. Power Syst. Clean Energy* **2022**, *10*, 805–817. [CrossRef]
9. Hu, J.; Vasilakos, A.V. Energy Big Data Analytics and Security: Challenges and Opportunities. *IEEE Trans. Smart Grid* **2016**, *7*, 2423–2436. [CrossRef]
10. Liu, J.; Weng, J.; Yang, A.; Chen, Y.; Lin, X. Enabling Efficient and Privacy-Preserving Aggregation Communication and Function Query for Fog Computing-Based Smart Grid. *IEEE Trans. Smart Grid* **2020**, *11*, 247–257. [CrossRef]
11. Liu, Z.; Cao, Z.; Dong, X.; Zhao, X.; Bao, H.; Shen, J. A verifiable privacy-preserving data collection scheme supporting multi-party computation in fog-based smart grid. *Front. Comput. Sci.* **2022**, *16*, 161810. [CrossRef]
12. Khan, H.M.; Khan, A.; Jabeen, F.; Rahman, A.U. Privacy preserving data aggregation with fault tolerance in fog-enabled smart grids. *Sustain. Cities Soc.* **2021**, *64*, 102522. [CrossRef]
13. Chen, S.; Yang, L.; Zhao, C.; Varadarajan, V.; Wang, K. Double-Blockchain Assisted Secure and Anonymous Data Aggregation for Fog-Enabled Smart Grid. *Engineering* **2022**, *8*, 159–169. [CrossRef]
14. Zhao, S.; Li, F.; Li, H.; Lu, R.; Ren, S.; Bao, H.; Lin, J.H.; Han, S. Smart and Practical Privacy-Preserving Data Aggregation for Fog-Based Smart Grids. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 521–536. [CrossRef]
15. Bonomi, F.; Milito, R.; Zhu, J.; Addepalli, S. Fog Computing and Its Role in the Internet of Things. In Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, Helsinki, Finland, 17 August 2012; pp. 13–16. [CrossRef]
16. Bonomi, F.; Milito, R.; Natarajan, P.; Zhu, J. Fog Computing: A Platform for Internet of Things and Analytics. In *Big Data and Internet of Things: A Roadmap for Smart Environments*; Springer: Cham, Switzerland, 2014; pp. 169–186. [CrossRef]
17. Dastjerdi, A.V.; Buyya, R. Fog Computing: Helping the Internet of Things Realize Its Potential. *Computer* **2016**, *49*, 112–116. [CrossRef]
18. Yang, M.; Zhu, T.; Liu, B.; Xiang, Y.; Zhou, W. Machine Learning Differential Privacy with Multifunctional Aggregation in a Fog Computing Architecture. *IEEE Access* **2018**, *6*, 17119–17129. [CrossRef]
19. Darzi, S.; Akhbari, B.; Khodaiemehr, H. LPM2DA: A Lattice-Based Privacy-Preserving Multi-Functional and Multi-Dimensional Data Aggregation Scheme for Smart Grid. *Clust. Comput.* **2022**, *25*, 263–278. [CrossRef]
20. Nyangaresi, V.O.; Abduljabbar, Z.A.; Mutlaq, K.A.A.; Ma, J.; Honi, D.G.; Aldarwish, A.J.Y.; Abduljaleel, I.Q. Energy Efficient Dynamic Symmetric Key Based Protocol for Secure Traffic Exchanges in Smart Homes. *Appl. Sci.* **2022**, *12*, 12688. [CrossRef]
21. Liang, G.; Weller, S.R.; Zhao, J.; Luo, F.; Dong, Z.Y. The 2015 Ukraine Blackout: Implications for False Data Injection Attacks. *IEEE Trans. Power Syst.* **2017**, *32*, 3317–3318. [CrossRef]
22. Xue, K.; Yang, Q.; Li, S.; Wei, D.S.L.; Peng, M.; Memon, I.; Hong, P. PPSO: A Privacy-Preserving Service Outsourcing Scheme for Real-Time Pricing Demand Response in Smart Grid. *IEEE Internet Things J.* **2019**, *6*, 2486–2496. [CrossRef]
23. Li, C.; Chen, Y.; Yang, Y.; Li, C.; Zeng, Y. PPCSB: A Privacy-Preserving Electricity Consumption Statistics and Billing Scheme in Smart Grid. In *Artificial Intelligence and Security, Proceedings of the 5th International Conference, New York, NY, USA, 26–28 July 2019*; Springer: New York, NY, USA, 2019; pp. 529–541. [CrossRef]
24. He, W. Real-time price scheme based on privacy protection. *Appl. Res. Comput.* **2019**, *36*, 1788–1792. [CrossRef]
25. Gope, P.; Sikdar, B. An Efficient Data Aggregation Scheme for Privacy-Friendly Dynamic Pricing-Based Billing and Demand-Response Management in Smart Grids. *IEEE Internet Things J.* **2018**, *5*, 3126–3135. [CrossRef]
26. Wang, X.; Liu, Y.; Choo, K.K.R. Fault-Tolerant Multisubset Aggregation Scheme for Smart Grid. *IEEE Trans. Ind. Inform.* **2021**, *17*, 4065–4072. [CrossRef]
27. Zuo, X.; Li, L.; Peng, H.; Luo, S.; Yang, Y. Privacy-Preserving Multidimensional Data Aggregation Scheme without Trusted Authority in Smart Grid. *IEEE Syst. J.* **2021**, *15*, 395–406. [CrossRef]
28. Xue, K.; Zhu, B.; Yang, Q.; Wei, D.S.L.; Guizani, M. An Efficient and Robust Data Aggregation Scheme without a Trusted Authority for Smart Grid. *IEEE Internet Things J.* **2020**, *7*, 1949–1959. [CrossRef]

29. Shen, H.; Liu, Y.; Xia, Z.; Zhang, M. An efficient aggregation scheme resisting on malicious data mining attacks for smart grid. *Inf. Sci.* **2020**, *526*, 289–300. [CrossRef]
30. Kumari, A.; Tanwar, S.; Tyagi, S.; Kumar, N.; Obaidat, M.S.; Rodrigues, J.J.P.C. Fog Computing for Smart Grid Systems in the 5G Environment: Challenges and Solutions. *IEEE Wirel. Commun.* **2019**, *26*, 47–53. [CrossRef]
31. Singh, A.K.; Kumar, J. A privacy-preserving multidimensional data aggregation scheme with secure query processing for smart grid. *J. Supercomput.* **2022**, 1–21. [CrossRef]
32. Zhang, X.; Tang, W.; Gu, D.; Zhang, Y.; Xue, J.; Wang, X. Lightweight Multidimensional Encrypted Data Aggregation Scheme with Fault Tolerance for Fog-Assisted Smart Grids. *IEEE Syst. J.* **2022**, *16*, 6647–6657. [CrossRef]
33. Braeken, A.; Kumar, P.; Martin, A. Efficient and Privacy-Preserving Data Aggregation and Dynamic Billing in Smart Grid Metering Networks. *Energies* **2018**, *11*, 2085. [CrossRef]
34. Chen, Z.; Zhang, J.; LI, Z. Design for secure two-party computation protocol based on ElGamal variant's homomorphic. *J. Commun.* **2015**, *36*, 204–211. [CrossRef]
35. Boneh, D.; Goh, E.J.; Nissim, K. Evaluating 2-DNF Formulas on Ciphertexts. In *Second International Conference on Theory of Cryptography*; Springer: Berlin/Heidelberg, Germany, 2005; pp. 325–341. [CrossRef]
36. Tsiounis, Y.; Yung, M. On the Security of ElGamal Based Encryption. In *Public Key Cryptography, First International Workshop on Practice and Theory in Public Key Cryptography, PKC '98, Pacifico Yokohama, Japan, 5–6 February 1998*; Lecture Notes in Computer Science; Springer: Berlin/Heidelberg, Germany, 1998; Volume 1431, pp. 117–134. [CrossRef]
37. Boneh, D.; Lynn, B.; Shacham, H. Short Signatures from the Weil Pairing. In Proceedings of the 7th International Conference on the Theory and Application of Cryptology and Information Security: Advances in Cryptology, Gold Coast, Australia, 9–13 December 2001; Springer: Berlin/Heidelberg, Germany, 2001; pp. 514–532. [CrossRef]