Hindawi
Security and Communication Networks
Volume 2020, Article ID 8845959, 14 pages
https://doi.org/10.1155/2020/8845959

Research Article

Privacy-Preserving Multidimensional Data Aggregation Scheme for Smart Grid

Yousheng Zhou,[1,2] **Xinyun Chen**,[1] **and Meihuan Chen**[1]

[1]*College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*
[2]*School of Cyber Security and Information Law, Chongqing University of Posts and Telecommunications, Chongqing 400065, China*

Correspondence should be addressed to Meihuan Chen; s170201055@stu.cqupt.edu.cn

Received 13 May 2020; Revised 23 October 2020; Accepted 16 November 2020; Published 3 December 2020

Academic Editor: Luxing Yang

In a smart grid, data aggregation is a common method to evaluate regional power consumption. Data leakage in the process of data transmission poses a security threat to the privacy of users. Many existing data aggregation schemes can only aggregate one-dimensional data; however, it is necessary to aggregate multidimensional data in practical smart grid applications. Therefore, this paper proposes a privacy-preserving multidimensional data aggregation scheme, which can aggregate multidimensional data and protect the individual user's identity and data privacy. The security of the proposed scheme is proved under the random oracle model. The simulation results show that the proposed scheme has great advantages in computing overhead, and the communication overhead also meets the requirements of the smart grid.

1. Introduction

A smart grid is a more efficient and modern grid. "Grid 2030" defines a smart grid as follows: "A fully automatic power transmission network that monitors and controls each user and node to ensure the two-way flow of power and information between power plants and power devices and all nodes between them" [1]. Smart grid consists of seven domains: the generation, transmission, distribution, customer, electricity market, service provider, and operation center domain, as shown in Figure 1.

Based on the real-time information of power consumption, the control center can monitor the power generation and consumption of each area, get the real-time power demand, and then take timely measures to optimize the power generation and distribution strategy. The customer can also get knowledge of current real-time power consumption and adjust his behavior to reduce expenses. In order to make a better transmission and distribution strategy, data aggregation is usually used to evaluate the power usage in a certain area. The purpose of data aggregation in a smart grid is to collect total power consumption data of users in a certain area and protect the power consumption data of an individual user from leakage.

In order to support various network functions, many smart devices such as smart terminals and smart meters have been deployed and used in the smart grid [2]. As wireless networks are increasingly used in smart grids, the communication channel between the smart meter and the control center may be open [3]. Therefore, the attacker can easily intercept, tamper with, or delete the messages in the communication channel, which causes great distress or economic loss. For example, an attacker can track a user's habits or lifestyle (he or she is at home or not at home) after obtaining his or her power consumption data, thus committing a crime inside a house [4]. An attacker may also cause economic loss to the user or service provider by injecting false information or making unreasonable demands.

With the wide application of the smart home, actual power data is possible to be multidimensional, for example, measuring power data by the type of household appliances

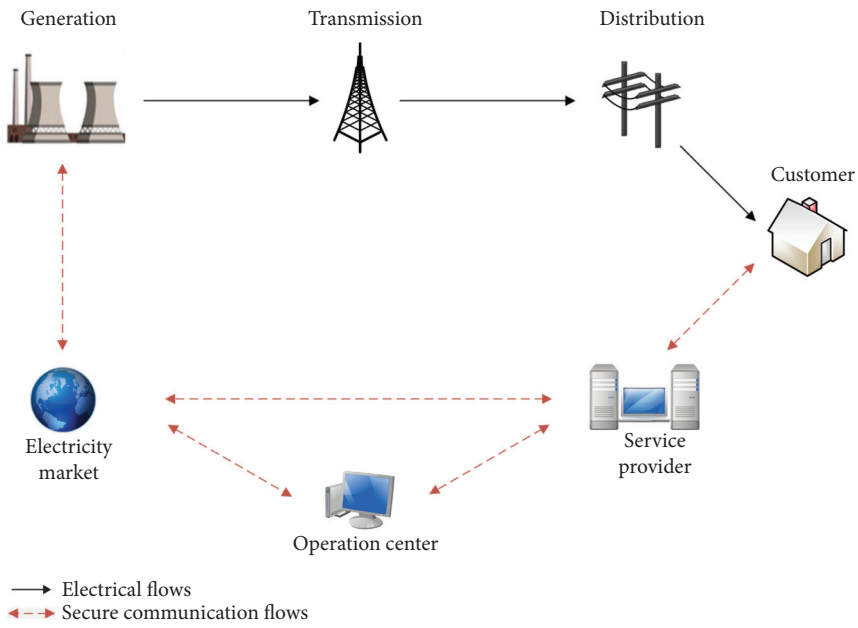

Figure 1: The network model for the smart grid.

such as refrigerators, air conditioners, and washing machines. Therefore, it is necessary to study multidimensional data aggregation in a smart grid.

In this paper, we propose a privacy-preserving multidimensional data aggregation scheme. The characteristics of the proposed scheme are as follows: (1) feasibility: the proposed scheme can aggregate multidimensional data; (2) security: the proposed scheme can protect the user's identity and data privacy; (3) robustness: the proposed scheme can work normally when any smart meter is off-line or out of order; (4) high efficiency: the proposed scheme adopts EC-ElGamal cryptosystem, and the computation performance is efficient.

The rest of this paper is organized as follows. In Section 2, we review the related works. In Section 3, we describe the preliminaries. In Section 4, we describe the system model and security requirements. We present our proposed scheme in Section 5. We analyze the security and performance in Sections 6 and 7, respectively. Finally, we make some conclusions in Section 8.

2. Related Work

Smart meters have three factors: smart meters' real-time power consumption data, smart meters' total power consumption data, and smart meters' identity [5]. In order to achieve the goal of data aggregation and protect the privacy of smart meters, several privacy-preserving one-dimensional data aggregation schemes have been proposed based on homomorphic encryption, blind factors, and Shamir's secret sharing.

Homomorphic encryption is a classical method for data aggregation. Homomorphic encryption can perform special algebraic operations on two or more ciphertexts to obtain the aggregated ciphertext, and the result of the decrypted aggregated ciphertext is the same as that of performing the same algebraic operation on the plaintext. At present, many privacy-preserving data aggregation schemes have been proposed based on homomorphic encryption (BGN cryptosystem [6], Paillier cryptosystem [7], EC-ElGamal cryptosystem [8], and lattice-based cryptosystem), such as BGN-based schemes [9–11], Paillier-based schemes [12–14], EC-ElGamal-based schemes [15–17], and lattice-based schemes [18].

Homomorphic encryption combined with blind factors (random numbers) is a common method to design data aggregation schemes. The trusted third party predistributes different blind factors to each user and the aggregator. Each user uses its own blind factor to obfuscate the power consumption data. When the aggregator receives data from all users, it can eliminate the blind factors added by all users to obtain aggregated data. Fan et al. [19] proposed the first data aggregation scheme that can resist internal attacks, which uses BGN cryptosystem and blind factors. Bao and Lu [20] found that Fan's scheme cannot provide data integrity. He et al. also proposed their scheme by using BGN cryptosystem and blind factors in [21], which can protect the integrity of data. The scheme proposed by He et al. [22] and Vahedi et al. [23] uses EC-ElGamal cryptosystem and blind factors and has high computational efficiency. All of the above [19–23] schemes can defend against internal attacks.

The combination of homomorphic encryption and Shamir's secret sharing is also used to design a data aggregation scheme [24]. For example, the PMDA scheme proposed by He [25] uses Shamir's secret sharing to allow smart meters to collectively negotiate aggregation parameters and supports multifunctional data aggregation. The 3PDA scheme proposed by Liu et al. [26] uses EC-ElGamal cryptosystem and Shamir's secret sharing, and users construct a virtual aggregation area to mask single data. The scheme presented in [25, 26] does not rely on a trusted third party. Even if any smart meter is off-line or out of order, the

system can work normally. New smart meters can be easily added to the system while the user's secret share remains the same.

At present, only one-dimensional data is considered in many schemes, but in practical application, power consumption data is usually multidimensional to facilitate fine-grained analysis. Based on the superincreasing sequence and Horner's rule, some researchers proposed multidimensional data aggregation schemes.

Knapsack cryptosystem based on superincreasing sequence can compress multidimensional data into one-dimensional data. The PPMA scheme proposed by Li et al. [27] uses Paillier cryptosystem and superincreasing sequence to aggregate multidimensional data. The PPMA scheme gives several successive power consumption ranges, divides the regional users into several subsets, and can get the sum of power consumption data of each subset and the number of users. The EPPA scheme proposed by Lu et al. [28] uses a superincreasing sequence to compress multidimensional data into one-dimensional data and then uses Paillier's cryptosystem to encrypt the compressed data.

The algorithm based on Horner's rule can also compress multidimensional data into one-dimensional data. The scheme proposed by Shen et al. [29] uses Paillier's cryptosystem and Horner's rule. Each user constructs Horner polynomial with the first Horner parameter, storing the multidimensional data in a single data. After embedding the second Horner parameter into the polynomial, Paillier's cryptosystem is used to encrypt the single data.

3. Preliminaries

3.1. Hard Problems. Let $\mathbb{G}$ be an additive cyclic group with prime order q; then some hard problems in the group $\mathbb{G}$ are described as follows.

Elliptic Curve Discrete Logarithm (DL) Problem. Given points $G, Q \in \mathbb{G}$, where $Q = aG$, $a \in Z_q^*$, a is unknown. DL problem is to compute the value of a.

Elliptic Curve Computational Diffie-Hellman (CDH) Problem. Given points $G, P, Q \in \mathbb{G}$, where $P = aG$ and $Q = bG$, $a, b \in Z_q^*$, a and b are unknown. CDH problem is to compute abG. CDH assumption holds if there exists no probabilistic polynomial-time adversary that can solve the CDH problem with a nonnegligible advantage.

Elliptic Curve Decisional Diffie-Hellman (DDH) Problem. Given a point $G, P, Q, Z \in \mathbb{G}$, where $P = aG$ and $Q = bG$, $a, b \in Z_q^*$, a and b are unknown. The DDH problem is to determine whether $Z = abG$ holds. DDH assumption holds if there exists no probabilistic polynomial-time adversary that can solve the DDH problem with a nonnegligible advantage.

3.2. Security Model of Authentication and Key Agreement. Define a probabilistic polynomial-time adversary $\mathcal{A}$, which can make a series of queries to simulate real attacks, define a simulator $\mathcal{S}$, and define a game played between $\mathcal{A}$ and $\mathcal{S}$. Then, $\mathcal{A}$ can adaptively make the following queries.

Hash(m): This query simulates an adversary's hash request for a message m. $\mathcal{S}$ needs to keep a table $L_H = (m, r)$. When $\mathcal{S}$ receives the request from $\mathcal{A}$, $\mathcal{S}$ checks if L_H contains a tuple (m, r). If so, $\mathcal{S}$ returns r to $\mathcal{A}$; otherwise, $\mathcal{S}$ randomly chooses r, stores (m, r) in L_H, and returns r to $\mathcal{A}$.

Execute(ID_i): $\mathcal{A}$ makes this query to simulate an eavesdropping attack (passive attack). $\mathcal{S}$ returns a copy of the exchange message executed under the real authentication protocol.

Send(ID_i, m): $\mathcal{A}$ makes this query to simulate an active attack. $\mathcal{A}$ can query the response information associated with the message m. $\mathcal{S}$ normally performs the steps of the authentication protocol and then returns the corresponding message to $\mathcal{A}$.

Corrupt(ID_i): $\mathcal{A}$ makes this query to simulate a corrosive attack that can obtain the participant's private key. $\mathcal{S}$ returns the relevant private key according to the authentication protocol.

Reveal(ID_i): $\mathcal{A}$ makes this query to simulate a known session key attack. If a valid session exists, $\mathcal{S}$ returns the session key corresponding to the participant; otherwise, $\mathcal{S}$ returns $\perp$.

Test(ID_i): This query simulates an adversary's ability to distinguish between a true session key and a random number. When the session key has been defined, $\mathcal{S}$ chooses a random number $b \in \{0, 1\}$. If $b = 1$, $\mathcal{S}$ returns the true session key to $\mathcal{A}$; if not, $\mathcal{S}$ returns a random number with the same length of the session key to $\mathcal{A}$.

After making the above queries, $\mathcal{A}$ can make *Test* query. The output of *Test* query depends only on the value of the bit b. The output of $\mathcal{A}$ is the result of guessing b' associated with the bit b. If $b = b'$, $\mathcal{A}$ wins the game. Define an event Succ as $\mathcal{A}$ wins the game. The advantage of $\mathcal{A}$ breaking the semantic security of the authentication protocol is

$$\mathrm{ADV}_A^{\mathrm{Protocol}} = |2\Pr[\mathrm{Succ}] - 1|. \quad (1)$$

Definition 1. The proposed authentication protocol is semantic secure if there exists no probabilistic polynomial-time adversary $\mathcal{A}$ that can win the above game with a nonnegligible advantage.

3.3. Security Model of Encryption and Signature. The encryption and signature scheme are used in this paper. Therefore, semantic security and unforgeability should be considered in the security model. Define a probabilistic polynomial-time adversary $\mathcal{A}$ that can make a series of queries to simulate real attacks, define a simulator $\mathcal{S}$, and define a game played between $\mathcal{A}$ and $\mathcal{S}$. Then $\mathcal{A}$ can adaptively make the following queries.

$h(m)$: This query simulates an adversary's request for a message m. $\mathcal{S}$ needs to keep a table $L_h = (m, r)$. When

$\mathcal{S}$ receives a request from $\mathcal{A}$, $\mathcal{S}$ checks if L_h contains a tuple (m, r). If so, $\mathcal{S}$ returns r to $\mathcal{A}$; otherwise, $\mathcal{S}$ randomly chooses r, stores (m, r) in L_h, and returns r to $\mathcal{A}$.

Creat(ID_i): This query simulates an adversary's attack to obtain the smart meter's public key. $\mathcal{S}$ needs to keep a table $L_U = (\mathrm{ID}_i, \mathrm{SK}, \mathrm{PK})$. When $\mathcal{S}$ receives the request from $\mathcal{A}$, $\mathcal{S}$ checks if table L_U exists in the public key PK with ID_i. If so, $\mathcal{S}$ returns PK to $\mathcal{A}$; otherwise, $\mathcal{S}$ randomly chooses a private key SK, generates the corresponding public key PK, stores $(\mathrm{ID}_i, \mathrm{SK}, \mathrm{PK})$ in L_U, and returns PK to $\mathcal{A}$.

Extract(ID_i): This query simulates an adversary's attack to obtain the smart meter's private key. $\mathcal{S}$ checks if table L_U exists in the private key SK with ID_i. If so, $\mathcal{S}$ returns SK to $\mathcal{A}$; otherwise, $\mathcal{S}$ randomly chooses a private key SK, generates the corresponding public key PK, stores $(\mathrm{ID}_i, \mathrm{SK}, \mathrm{PK})$ in L_U, and returns SK to $\mathcal{A}$

Encrypt(ID_i, m_i): this query simulates an adversary's encryption request for a message m_i. $\mathcal{S}$ queries the public key PK with ID_i, uses PK to encrypt the message m_i, and then returns the ciphertext to $\mathcal{A}$.

Sign(ID_i, m_i): This query simulates an adversary's signature request for a message m_i. $\mathcal{S}$ queries the private key SK with ID_i, uses SK to sign message m_i, and then returns the message m_i and signature to $\mathcal{A}$.

Unsign(ID_i, mi): This query simulates an adversary's request to verify the message m_i's signature. $\mathcal{S}$ queries the public key PK with ID_i and uses PK to validate the signature of the message m_i.

Definition 2. If there exists no probabilistic polynomial-time adversary that can win the following game with a nonnegligible advantage, the proposed scheme is secure against indistinguishability under the chosen-plaintext attack (IND-CPA).

Initialization: $\mathcal{S}$ runs a key generation algorithm, generates a key pair $(\mathrm{PK}, \mathrm{SK})$, sends the public key PK to $\mathcal{A}$, and keeps the private key SK

Phase 1: $\mathcal{A}$ can access a random oracle to make a series of queries. $\mathcal{A}$ randomly chooses two plaintexts m_0, m_1 with the same length and sends them to $\mathcal{S}$.

Challenge: $\mathcal{S}$ randomly chooses a bit $u \in \{0, 1\}$ and sends the ciphertext C_u of the message m_u to $\mathcal{A}$. We call the ciphertext C_u the challenging ciphertext

Guess: $\mathcal{A}$ outputs its guess $u' \in \{0, 1\}$. The advantage of $\mathcal{A}$ in the above game is defined as follows:

$$\mathrm{ADV}_A^{\mathrm{IND-CPA}} = \left|2\Pr\left[u = u'\right] - 1\right|. \quad (2)$$

Definition 3. If there exists no probabilistic polynomial-time adversary that can win the following game with a non-negligible advantage, the proposed scheme is secure against existential unforgeability under the adaptive chosen messages attacks (EUF-CMA).

Initialization: $\mathcal{S}$ runs a key generation algorithm, generates a key pair $(\mathrm{PK}, \mathrm{SK})$, sends the public key PK to $\mathcal{A}$ (also known as a forger), and keeps the private key SK.

Query 1: $\mathcal{A}$ queries the hash value of the message m, and $\mathcal{S}$ returns the corresponding hash value $h(m)$ to $\mathcal{A}$.

Query 2: $\mathcal{A}$ queries the signature of the message m, and $\mathcal{S}$ returns the corresponding signature Sign(m) to $\mathcal{A}$.

Challenge: $\mathcal{A}$ forges a message's signature pair $(m^*, \mathrm{Sign}(m^*))$ and sends it to $\mathcal{S}$. $\mathcal{S}$ verifies the validity of the signature. If the forged signature is valid, $\mathcal{A}$ succeeds; otherwise, $\mathcal{A}$ fails.

4. System Model and Security Requirement

4.1. System Model. In the proposed scheme, the system model for the smart grid consists of four entities: smart meter (SM), aggregator (AGG), control center (CC), and trusted third party (TTP), as shown in Figure 2.

SM: It is responsible for regularly collecting real-time power consumption data of the user and sending encrypted data to the aggregator. The smart meter is honest-but-curious. It operates according to the protocol and may infer information from other users.

AGG: It is responsible for aggregating the power consumption data of all users and sending the aggregated data to the control center. The aggregator is honest-but-curious. It stores all intermediate computational results and may get users' privacy information from them

CC: It is responsible for decrypting and analyzing aggregated data to obtain the sum of users' power consumption data for each area and to generate an appropriate response. The control center is fully trusted and may attempt to analyze incoming messages to obtain valuable information.

TTP: It is responsible for generating and distributing security parameters for all smart meters. TTP is fully trusted and participate only in the registration process, not in the data aggregation process.

4.2. Security Requirement. According to related works in recent years, the data aggregation scheme in a smart grid should meet three security requirements. We summarize these requirements as follows.

Confidentiality: A malicious attacker may intercept information from a user. The leakage of a user's power consumption data can compromise its privacy. Therefore, it is important to ensure that the attacker cannot obtain the power consumption data of an individual user.

Integrity: A malicious attacker may tamper with a message sent by a user, which will affect the normal statistical analysis. Therefore, it is important to ensure that the messages sent by the user are correct.

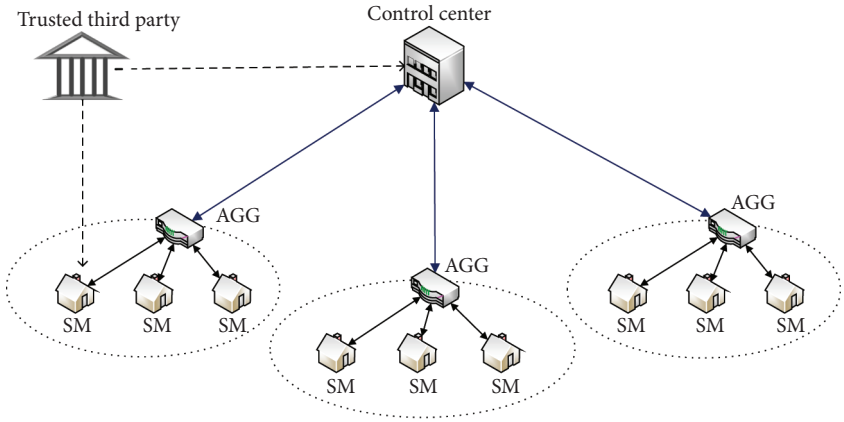

Figure 2: The system model of the proposed scheme.

Authentication: A malicious attacker may forge a message and impersonate a real user to send a message, which will affect the normal process of statistical analysis. Therefore, it is important to ensure that the data received by the aggregator is from a legitimate user.

5. Scheme Construction

This section describes the proposed privacy-preserving multidimensional data aggregation scheme. The proposed scheme consists of six steps: system setup, registration and login, authentication and key agreement, data generation, data aggregation, and multidimensional data decryption. We assume that there are n users in each residential area. The symbols and their definitions used in this section are shown in Table 1.

5.1. System Setup. The initialization phase is used to generate system public parameters. The smart meter, aggregator, control center, and trusted third party randomly choose an integer from Z_q^* as their private key and compute the corresponding public key $X_i = x_i G$, $X_A = x_A G$, $X_{CC} = x_{CC} G$, $X_T = x_T G$.

5.2. Registration and Login

(1) *Registration.* All smart meters need to register, and each smart meter only needs to register once. The detailed steps for the registration phase are described as follows:

  (1) SM_i submits its identity ID_i and password PW_i to TTP through a secure channel.
  (2) After receiving the message $\{ID_i, PW_i\}$, TTP saves the identity and password information and computes $A_i = h(PW_i \| x_T)$, $PID_i = ID_i \oplus A_i$, and $B_i = h(ID_i \| PW_i \| A_i)$. Finally, TTP returns the message $\{PID_i, B_i\}$ to SM_i through a secure channel.
  (3) After receiving the message $\{PID_i, B_i\}$, SM_i saves $\{PID_i, B_i\}$ in its own memory (with some tamper-proof ability).

(2) *Login.* SM_i needs to perform a login phase before communicating with the aggregator. The detailed steps of the login phase are described as follows:

  (1) SM_i computes $A_i = ID_i \oplus PID_i$ and $B_i' = h(ID_i \| PW_i \| A_i)$
  (2) SM_i verifies whether the parameter B_i stored in memory is equal to B_i'. If so, the login of SM_i succeeds; otherwise, the login of SM_i fails

5.3. Authentication and Key Agreement. The goal of the authentication and key agreement phase is for the smart meter to request authentication from the aggregator and establish a session key between the smart meter and the aggregator, as shown in Figure 3. The session key is used by the aggregator to encrypt the response message using a symmetric encryption algorithm when the response message is returned. The detailed steps for authentication and key agreement phase are described as follows.

(1) SM_i randomly chooses an integer $r_i \in Z_q^*$ and computes $AID_i = ID_i \oplus H_0(r_i X_A)$, $R_i = r_i G$, and $\delta_1 = H_1(AID_i \| ID_i \| R_i \| t_i)$, where t_i is the current timestamp. Finally, SM_i sends the message $\{AID_i, R_i, \delta_1, t_i\}$ to AGG

(2) After receiving the message $\{AID_i, R_i, \delta_1, t_i\}$, AGG computes $ID_i = AID_i \oplus H_0(x_A R_i)$ and checks whether the equation $\delta_1 = H_1(AID_i \| ID_i \| R_i \| t_i)$ holds. If not, AGG terminates the communication; otherwise, AGG randomly chooses $r_j \in Z_q^*$ and computes $T_j = r_j G$, $S_{ij} = r_j R_i$, $sk_{ij} = H_1(R_i \| T_j \| S_{ij} \| ID_i)$, and $\delta_2 = H_1(sk_{ij} \| ID_i \| t_j)$, where sk_{ij} is the session key and t_j is the current timestamp. Finally, AGG returns the message $\{T_j, \delta_2, t_j\}$ to SM_i

(3) After receiving the message $\{T_j, \delta_2, t_j\}$, SM_i computes $S_{ij} = r_i T_j$, $sk_{ij} = H_1(R_i \| T_j \| sk_{ij} \| ID_i)$ and checks whether the equation $\delta_2 = H_1(sk_{ij} \| ID_i \| t_j)$

Table 1: Symbols and definitions.

Symbol	Definition
SM_i	The i – th smart meter
AGG	The aggregator (usually a gateway)
q	a large prime
$\mathbb{G}$	An elliptic curve group of order q
H_0	A one-way hash function H_0: $\mathbb{G} \longrightarrow Z_q^*$
h, H_1, H_2, H_3, H_4	A one-way hash function h, H_1, H_2, H_3, H_4: $\{0,1\}^* \longrightarrow Z_q^*$
ID_i	The identity of SM_i
AID_i	The pseudonym of SM_i
(x_i, X_i)	The private key and public key of SM_i
(x_A, X_A)	The private key and public key of AGG
(x_{CC}, X_{CC})	The private key and public key of CC
(x_T, X_T)	The private key and public key of TTP
	The connection symbol

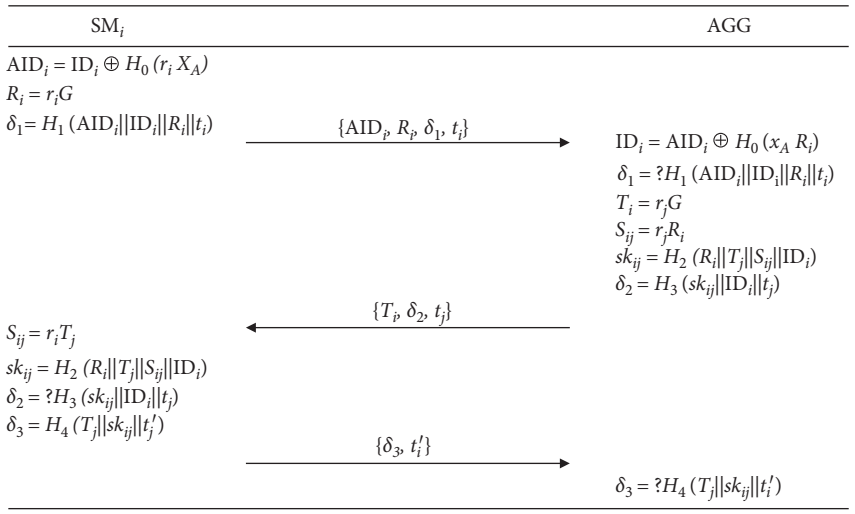

Figure 3: The authentication and key agreement phase of the proposed scheme.

holds. If not, SM_i terminates the communication; otherwise, SM_i computes $\delta_3 = H_1(T_j \| sk_{ij} \| t_i')$, where sk_{ij} is the session key and t_i' is the current timestamp. Finally, SM_i sends the message $\{\delta_3, t_i'\}$ to AGG

(4) AGG verifies whether $\delta_3 = H_1(T_j \| sk_{ij} \| t_i')$ holds. If so, the authentication of SM_i succeeds; otherwise, the authentication of SM_i fails

5.4. Data Generation. Assume that SM_i can obtain l-dimensional power consumption data $\{d_{i1}, d_{i2}, \ldots, d_{il}\}$. The detailed steps for generating the ciphertext are described as follows.

(1) SM_i randomly chooses $s_i \in Z_q^*$ and computes the ciphertext according to

$$C_i = \left(s_i G, d_{i1} G + s_i \left(X_A + X_{CC}\right), d_{i2} G + s_i \left(X_A + X_{CC}\right), \ldots, d_{il} G + s_i \left(X_A + X_{CC}\right)\right). \quad (3)$$

(2) SM_i randomly chooses $k_i \in Z_q^*$ and computes its signature (K_i, z_i) according to

$$\left(K_i = k_i G, \phi_i = h\left(AID_i \| X_i \| C_i \| K_i \| T_i\right), z_i = k_i + \phi_i \cdot x_i \bmod q\right). \quad (4)$$

(3) SM_i sends the message $AID_i \| X_i \| C_i \| K_i \| T_i \| z_i$ to AGG, where T_i is the current timestamp.

5.5. Data Aggregation. After receiving the message $AID_i \| X_i \| C_i \| K_i \| T_i \| z_i$, AGG verifies the smart meter's signature and computes the aggregated ciphertext, as shown in Figure 4. Suppose that there are currently k smart meters participating in the data aggregation.

(1) AGG verifies signatures of all smart meters according to

$$\sum_{i=1}^{k} z_i \cdot G = ? \sum_{i=1}^{k} K_i + \sum_{i=1}^{k} h\left(AID_i \| X_i \| C_i \| K_i \| T_i\right) \cdot X_i. \quad (5)$$

(2) AGG computes the aggregated ciphertext C according to

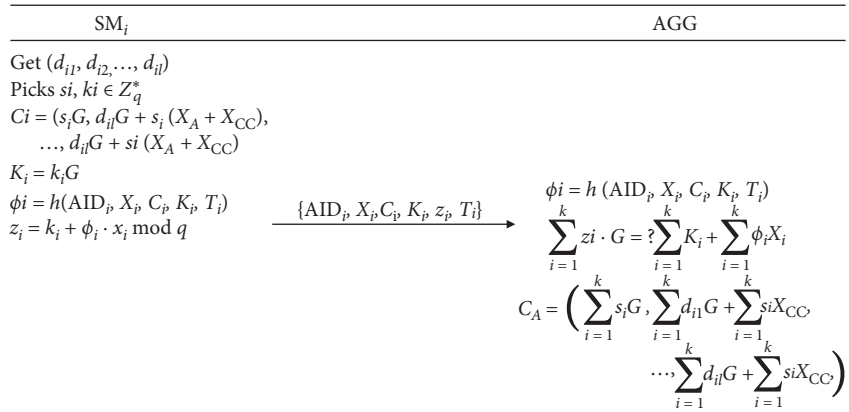

Figure 4: The data aggregation phase of the proposed scheme.

$$C = \sum_{i=1}^{k} C_i = \left(\sum_{i=1}^{k} s_i G, \sum_{i=1}^{k} d_{i1} G + \sum_{i=1}^{k} s_i \left(X_A + X_{\mathrm{CC}} \right), \sum_{i=1}^{k} d_{i2} G + \sum_{i=1}^{k} s_i \left(X_A + X_{\mathrm{CC}} \right), \ldots, \sum_{i=1}^{k} d_{il} G + \sum_{i=1}^{k} s_i \left(X_A + X_{\mathrm{CC}} \right) \right). \tag{6}$$

(3) AGG partially decrypts the aggregated ciphertext C using the private key x_A according to

$$\begin{aligned} &\left(\sum_{i=1}^{k} d_{i1} G + \sum_{i=1}^{k} s_i \left(X_A + X_{\mathrm{CC}} \right) - x_A \sum_{i=1}^{k} s_i G, \sum_{i=1}^{k} d_{i2} G + \sum_{i=1}^{k} s_i \left(X_A + X_{\mathrm{CC}} \right) - x_A \sum_{i=1}^{k} s_i G, \ldots, \sum_{i=1}^{k} d_{il} G + \sum_{i=1}^{k} s_i \left(X_A + X_{\mathrm{CC}} \right) - x_A \sum_{i=1}^{k} s_i G \right) \\ &= \left(\sum_{i=1}^{k} d_{i1} G + \sum_{i=1}^{k} s_i X_{\mathrm{CC}}, \sum_{i=1}^{k} d_{i2} G + \sum_{i=1}^{k} s_i X_{\mathrm{CC}}, \ldots, \sum_{i=1}^{k} d_{il} G + \sum_{i=1}^{k} s_i X_{\mathrm{CC}} \right). \end{aligned} \tag{7}$$

Therefore, the form of partially decrypted ciphertext C_A is shown in

$$C_A = \left(\sum_{i=1}^{k} s_i G, \sum_{i=1}^{k} d_{i1} G + \sum_{i=1}^{k} s_i X_{\mathrm{CC}}, \sum_{i=1}^{k} d_{i2} G + \sum_{i=1}^{k} s_i X_{\mathrm{CC}}, \ldots, \sum_{i=1}^{k} d_{il} G + \sum_{i=1}^{k} s_i X_{\mathrm{CC}} \right). \tag{8}$$

(4) AGG randomly chooses $k_A \in Z_q^*$ and computes its signature (K_A, z_A) according to

$$\left(K_A = k_A G, \phi_A = h\left(ID_A \| X_A \| C_A \| K_A \| T_A \right), z_A = k_A + \phi_A \cdot x_A \right). \tag{9}$$

(5) AGG sends the message $\mathrm{ID}_A \| X_A \| C_A \| K_A \| T_A \| z_A$ to CC, where T_A is the current timestamp.

5.6. Data Decryption. After receiving the message $\mathrm{ID}_A \| X_A \| C_A \| K_A \| T_A \| z_A$, CC verifies the signature of AGG. After successful verification, CC decrypts the aggregated ciphertext C_A using its own private key x_{CC} to get the sum of the power consumption data.

(1) CC verifies the signature of AGG according to

$$z_A \cdot G = ? K_A + h\left(\mathrm{ID}_A \| X_A \| C_A \| K_A \| T_A \right) \cdot X_A. \tag{10}$$

(2) CC decrypts the aggregated ciphertext C_A using the private key x_{CC} according to

$$\begin{aligned}&\left(\sum_{i=1}^{k} d_{i1}G + \sum_{i=1}^{k} s_i X_{CC} - x_{CC}\sum_{i=1}^{k} s_i G,\right.\\&\sum_{i=1}^{k} d_{i2}G + \sum_{i=1}^{k} s_i X_{CC} - x_{CC}\sum_{i=1}^{k} s_i G, \ldots,\\&\left.\sum_{i=1}^{k} d_{il}G + \sum_{i=1}^{k} s_i X_{CC} - x_{CC}\sum_{i=1}^{k} s_i G\right)\\&= \left(\sum_{i=1}^{k} d_{i1}G, \sum_{i=1}^{k} d_{i2}G, \ldots, \sum_{i=1}^{k} d_{il}G\right).\end{aligned} \tag{11}$$

(3) Using Pollard's lambda algorithm, the sum of the power consumption data for each dimension can be computed, as shown in (12), where $\sum_{i=1}^{k} d_{il}$ represents the total power consumption data of k users in the l – th dimension.

$$\begin{aligned}&\left(\sum_{i=1}^{k} d_{i1}G, \sum_{i=1}^{k} d_{i2}G, \ldots, \sum_{i=1}^{k} d_{il}G\right)\\&= \left(\log_G \sum_{i=1}^{k} d_{i1}, \log_G \sum_{i=1}^{k} d_{i2}, \ldots, \log_G \sum_{i=1}^{k} d_{il}\right).\end{aligned} \tag{12}$$

6. Security Analysis

6.1. Formal Security Analysis

Theorem 1. *Assume that $ADV_A^{Ptotocol}$ represents the advantage of a probabilistic polynomial-time adversary to break the semantic security of the proposed authentication protocol; then,*

$$\begin{aligned}\mathrm{ADV}_A^{\mathrm{Ptotocol}} \le{}& \frac{2(q_{\mathrm{send}} + q_{\mathrm{exe}})}{L} + \frac{q_{H_0}^2}{|H_0|} + \frac{q_{H_1}^2}{|H_1|} + \frac{q_{H_2}^2}{|H_2|}\\&+ \frac{q_{H_3}^2}{|H_3|} + \frac{4}{q} + 2\mathrm{ADV}_G^{\mathrm{CDH}}(t),\end{aligned} \tag{13}$$

where L represents the size of the identity space, $|H_i|$ represents the size of the hash function space, q is the prime order of group $\mathbb{G}$, and q_{send}, q_{exe}, and q_{H_i} represent the number of Send query, Execute query, and Hash query, respectively.

Proof. Define a series of games $\mathrm{Game}_0, \mathrm{Game}_1, \mathrm{Game}_2$, and Game_6. We use $\mathrm{Succ}_j\,(j = 0, 1, \ldots, 6)$ to indicate the event that A successfully guesses $b = b'$ in *Test* query in Game_j.

Game_0: this game simulates real-world attacks by an adversary. The value of b is chosen at random. Therefore, according to the authentication and key agreement model, we have

$$\mathrm{ADV}_A^{\mathrm{Protocol}} = 2\left|\Pr[\mathrm{Succ}_0] - 1\right|. \tag{14}$$

Game_1: this game uses Execute, Send, Reveal, Corrupt, and Test query to simulate real attacks. Game_0 and Game_1 are indistinguishable. Therefore, we have

$$\Pr[\mathrm{Succ}_1] = \Pr[\mathrm{Succ}_0]. \tag{15}$$

Game_2: this game simulates all queries in Game_1; the only difference is that Game_2 will simulate an adversary's guess attack on the smart meter's true identity. Since the smart meter's identity is converted to a pseudonym by a random number during each authentication phase, the adversary is unable to determine the smart meter's true identity and has no other information to verify the smart meter's true identity. Therefore, we have

$$\left|\Pr[\mathrm{Succ}_1] - \Pr[\mathrm{Succ}_2]\right| \le \frac{q_{\mathrm{send}} + q_{\mathrm{exe}}}{L}. \tag{16}$$

Game_3: this game simulates all queries in Game_2; the only difference is that Game_3 will simulate collision attacks that occur on messages $\{\mathrm{AID}_i, R_i, \delta_i, t_i\}, \{T_j, \delta_2, t_j\}$, and $\{\delta_3, t_i'\}$. Therefore, we have

$$\left|\Pr[\mathrm{Succ}_2] - \Pr[\mathrm{Succ}_3]\right| \le \frac{q_{H_0}^2}{2|H_0|} + \frac{q_{H_1}^2}{2|H_1|} + \frac{q_{H_2}^2}{2|H_2|} + \frac{q_{H_3}^2}{2|H_3|}. \tag{17}$$

Game_4: this game simulates all queries in Game_3; the only difference is that Game_4 will simulate the adversary's corrosion attack on the participant. When Corrupt query is executed, the private key stored in the smart meter and in the aggregator can be extracted by the adversary. However, this information is useless for calculating the session key, because a secret random number that is generated temporarily must be required. Due to the fact that r_i and r_j are randomly selected from Z_q^*, we have

$$\left|\Pr[\mathrm{Succ}_3] - \Pr[\mathrm{Succ}_4]\right| \le \frac{2}{q}. \tag{18}$$

Game_5: this game simulates all queries in Game_4; the only difference is that other hash functions will be used to compute the temporary session key sk_{ij}. That is, instead of using a random oracle, we use $\mathrm{sk}_{ij} = \mathrm{Hash}'(R_i \| T_j \| S_{ij} \| \mathrm{ID}_i)$ to generate the session key. Game_5 and Game_4 are indistinguishable unless an event $\mathrm{Ask}\,H_5(R_i \| T_j \| S_{ij} \| \mathrm{ID}_i)$ occurs. $\mathrm{Ask}H_5(R_i \| T_j \| S_{ij} \| \mathrm{ID}_i)$ represents the adversary that makes a query about message $R_i \| T_j \| S_{ij} \| \mathrm{ID}_i$ to the random oracle. No matter how many Test queries are made, all results are independently random. Therefore, we have

$$\left|\Pr[\mathrm{Succ}_5]\right| \le \frac{1}{2},$$
$$\left|\Pr[\mathrm{Succ}_4] - \Pr[\mathrm{Succ}_5]\right| \le \Pr[\mathrm{Ask}H_5]. \tag{19}$$

Game_6: this game simulates all queries in Game_5; the only difference is that Game_6 will simulate an event where the adversary breaks CDH problem, randomly choose two integers $a, b \in Z_q^*$, given an instance of CDH problem (aG, bG), and compute $A = aG$, $B = bG$; then, we have $\mathrm{Ask}H_6(R_i \| T_j \| \mathrm{CDH}(aG, bG) \| \mathrm{ID}_i)$. In Game_5, the adversary needs to make a query such as $\mathrm{Ask}H_5(R_i \| T_j \| Z \| \mathrm{ID}_i)$, where $Z = \mathrm{CDH}(A, B)$. Therefore, we have $\Pr[\mathrm{Ask}H_5] = \Pr[\mathrm{Ask}H_6] = \Pr[\mathrm{Succ}_6]$. In addition, we have

$$\left|\Pr[\mathrm{Succ}_6]\right| \le \mathrm{ADV}_G^{\mathrm{CDH}}(t), \tag{20}$$

where $\mathrm{ADV}_G^{\mathrm{CDH}}(t)$ represents the advantage of $\mathscr{A}$ to break the CDH problem.

From the above analysis, we have

$$\begin{aligned}\left|\Pr[\mathrm{Succ}_0] - \frac{1}{2}\right| &= \left|\Pr[\mathrm{Succ}_0] - \Pr[\mathrm{Succ}_5]\right| \\ &\le \left|\Pr[\mathrm{Succ}_0] - \Pr[\mathrm{Succ}_1]\right| + \left|\Pr[\mathrm{Succ}_1] - \Pr[\mathrm{Succ}_2]\right| + \left|\Pr[\mathrm{Succ}_2] - \Pr[\mathrm{Succ}_3]\right| + \\ &\quad \cdot \left|\Pr[\mathrm{Succ}_3] - \Pr[\mathrm{Succ}_4]\right| + \left|\Pr[\mathrm{Succ}_4] - \Pr[\mathrm{Succ}_5]\right| \\ &\le \frac{q_{\mathrm{send}} + q_{\mathrm{exe}}}{L} + \frac{q_{H_0}^2}{2|H_0|} + \frac{q_{H_1}^2}{2|H_1|} + \frac{q_{H_2}^2}{2|H_2|} + \frac{q_{H_3}^2}{2|H_3|} + \frac{2}{q} + \mathrm{ADV}_G^{\mathrm{CDH}}.\end{aligned} \tag{21}$$

Due to $\mathrm{ADV}_A^{\mathrm{Protocol}} \le |2\Pr[\mathrm{Succ}_0] - 1|$, we have

$$\begin{aligned}\mathrm{ADV}_A^{\mathrm{Protocol}} \le{}& \frac{2(q_{\mathrm{send}} + q_{\mathrm{exe}})}{L} + \frac{q_{H_0}^2}{|H_0|} + \frac{q_{H_1}^2}{|H_1|} + \frac{q_{H_2}^2}{|H_2|} \\ &+ \frac{q_{H_3}^2}{|H_3|} + \frac{4}{q} + 2\mathrm{ADV}_G^{\mathrm{CDH}}(t).\end{aligned} \tag{22}$$

To sum up, the advantage of adversary $\mathscr{A}$ to break the proposed authentication protocol is negligible, and the proposed authentication protocol is semantic secure.

Theorem 2. *The proposed scheme is secure against IND-CPA, if the DDH problem is hard.*

Proof. Assume that there exists a probabilistic polynomial-time adversary $\mathscr{A}$ that can win the game in Definition 2 with a nonnegligible advantage ε. Then, we can construct a simulator $\mathscr{S}$ to solve DDH problem with a nonnegligible advantage ε'. The simulator $\mathscr{S}$ chooses a challenging identity ID_I, and the adversary $\mathscr{A}$ can make the following queries:

$H_i(m)$: $\mathscr{S}$ needs to keep a table $L_{H_i} = (m, r)$, where $i = 0, 1, 2, 3, 4$. After receiving the hash request from $\mathscr{A}$, $\mathscr{S}$ checks if the tuple (m, r) exists in L_{H_i}. If so, $\mathscr{S}$ returns r to $\mathscr{A}$; otherwise, $\mathscr{S}$ randomly chooses r, stores (m, r) into L_{H_i}, and returns r to $\mathscr{A}$

$h(\mathrm{ID}_i, X_i, C_i, K_i, T_i)$: $\mathscr{S}$ needs to keep a table $L_h = (\mathrm{ID}_i, X_i, C_i, K_i, T_i, r)$. After receiving the hash request from $\mathscr{A}$, $\mathscr{S}$ checks if $(\mathrm{ID}_i, X_i, C_i, K_i, T_i, r)$ exists in L_h. If so, $\mathscr{S}$ returns r to $\mathscr{A}$; otherwise, $\mathscr{S}$ randomly chooses $r \in Z_q^*$, stores $(\mathrm{ID}_i, X_i, C_i, K_i, T_i, r)$ into L_h, and returns r to $\mathscr{A}$

Creat(ID_i): $\mathscr{S}$ needs to keep a table $L_U = (\mathrm{ID}_i, x_i, X_i)$. After receiving the request from $\mathscr{A}$, $\mathscr{S}$ checks if $(\mathrm{ID}_i, x_i, X_i)$ exists in L_U. If so, $\mathscr{S}$ returns X_i to $\mathscr{A}$; otherwise, $\mathscr{S}$ randomly chooses $x_i \in Z_q^*$, computes $X_i = x_i G$, stores $(\mathrm{ID}_i, x_i, X_i)$ into L_U, and returns X_i to $\mathscr{A}$

Extract(ID_i): after receiving the request from $\mathscr{A}$, $\mathscr{S}$ first checks whether the identity ID_i used by $\mathscr{A}$ in the query is equivalent to the challenging identity ID_I. If so, $\mathscr{S}$ terminates this game; otherwise, $\mathscr{S}$ checks if $(\mathrm{ID}_i, x_i, X_i)$ exists in L_U. If so, $\mathscr{S}$ returns x_i to $\mathscr{A}$; otherwise, $\mathscr{S}$ makes Creat(ID_i) query to generate the private key and public key x_i and X_i, stores $(\mathrm{ID}_i, x_i, X_i)$ into L_U, and returns x_i to $\mathscr{A}$

Encrypt(ID_i, m_i): after receiving the request from $\mathscr{A}$, $\mathscr{S}$ checks if $(\mathrm{ID}_i, x_i, X_i)$ exists in L_U. If so, $\mathscr{S}$ uses X_i to generate the ciphertext; otherwise, $\mathscr{S}$ makes Creat(ID_i) query to generate the private key and public key x_i and X_i and then uses X_i to generate the ciphertext □

Proof. Assume that the ciphertext $C_i = m_i G + s_i X$ is secure against IND-CPA. Define a series of games $\mathrm{Game}_0, \mathrm{Game}_1$, and Game_2. With these games, we reduce the instance of the DDH problem. That is, given $(G, P = aG, Q = bG, Z)$, determine whether $Z = abG$, where $G, P, Q, Z \in \mathbb{G}$, $a, b \in Z_q^*$, and a and b are unknown. $\mathscr{S}$ chooses a challenging identity ID_I.

Game_0: this game simulates real-world attacks. $\mathscr{S}$ acts as a smart meter, knowing the public and private key pair $(S_i = s_i G, s_i)$. $\mathscr{A}$ knows the public key and has access to the random oracle. At some point, $\mathscr{A}$ randomly chooses an identity ID_i and two plaintexts

(m_{i0}, m_{i1}) with the same length and sends them to $\mathcal{S}$ for an encryption query. Then, $\mathcal{S}$ chooses a bit $u \in \{0,1\}$, encrypts the ciphertext $C_i = m_{iu}G + s_iX$, and sends the ciphertext C_i to $\mathcal{A}$. Finally, $\mathcal{A}$ outputs its guess $u' \in \{0,1\}$. Succ_0 represents the event in Game_0 that $u = u'$, and we use symbols Succ_j $(j = 0,1,2)$ to represent the same meaning in any game. Based on Definition 2, we have

$$\varepsilon = \left|2\Pr[u = u'] - 1\right|. \tag{23}$$

Game_1: in this game, we embed the instance (G, aG, bG, Z) of the DDH problem. When $\mathcal{A}$ makes $\text{Creat}(\text{ID}_i)$ query, $\mathcal{S}$ randomly chooses $r_i \in Z_q^*$, sets $X = r_ibG$, saves (ID_i, r_i, X) into L_U, and sends X to $\mathcal{A}$. Because $X = r_ibG$ is evenly distributed in the group $\mathbb{G}$, Game_1 is completely indistinguishable from Game_0. Therefore, we have

$$\Pr[\text{Succ}_1] = \Pr[\text{Succ}_0]. \tag{24}$$

Game_2: in this game, $\mathcal{S}$ replaces the public key $S_i = s_iG$ with $S_i = aG$. $\mathcal{S}$ does not know the private key a. Therefore, when $\mathcal{A}$ makes $\text{Encrypt}(\text{ID}_i, m_i)$ query, $\mathcal{S}$ performs the following steps. (1) When $\text{ID}_i = \text{ID}_I$, $\mathcal{S}$ looks for the record (ID_I, r_I, X) in L_U. (2) $\mathcal{S}$ computes $C_I = m_{Iu}G + r_IZ$ and sends C_I to $\mathcal{A}$. (3) Define $Z = abG \in \mathbb{G}$ as event E.

If event E actually occurs, then C_I is a valid ciphertext when public key $S_i = aG$ and $X = r_IbG$ holds. Therefore, at this time, $\mathcal{A}$ can play its ability to guess whether $u = u'$.

However, if event E does not occur, $\mathcal{A}$ can only guess $u = u'$ at a random probability of $(1/2)$. Therefore, we have

$$\Pr[\text{Succ}_2|\overline{E}] = \frac{1}{2}. \tag{25}$$

Therefore, based on the above analysis, we can solve the DDH problem with probability ε'.

$$\begin{aligned}
\varepsilon' &= \left|\Pr[\text{Succ}_2|E] - \Pr[\text{Succ}_2|\overline{E}]\right| \\
&= \left|\Pr[\text{Succ}_1] - \frac{1}{2}\right| \\
&= \left|\Pr[\text{Succ}_0] - \frac{1}{2}\right| \\
&= \left|\Pr[u = u'] - \frac{1}{2}\right|.
\end{aligned} \tag{26}$$

Due to $\varepsilon = |2\Pr[u = u'] - 1|$, $\varepsilon' = |\Pr[u = u'] - (1/2)| = (\varepsilon/2)$.

Because the advantage ε in the previous assumption cannot be ignored, ε' cannot be ignored. That is, a simulator $\mathcal{S}$ can be constructed to solve the DDH problem. However, DDH problem cannot be solved in practice; then, the conclusion is impossible. Therefore, our assumption does not hold. In other words, the proposed scheme is secure against indistinguishability under the chosen-plaintext attack (IND-CPA). □

Theorem 3. *The proposed scheme is secure against EUF-CMA, if the discrete logarithm problem is hard.*

Proof. Assume that there exists a probabilistic polynomial-time adversary that can win the game in Definition 3 with a nonnegligible advantage ε. Then, we can construct a simulator $\mathcal{S}$ to solve the discrete logarithm problem. Given an instance (P, Q) of a discrete logarithm problem, where $P, Q \in \mathbb{G}$, the goal of $\mathcal{S}$ is to find $x \in Z_q^*$, such that $Q = xG$. $\mathcal{S}$ chooses a challenging identity ID_I. $\mathcal{A}$ can make $H_i(m)$, $h(\text{ID}_i, X_i, C_i, K_i, T_i)$, $\text{Extract}(\text{ID}_i)$ queries as it did in Theorem 2. The adversary can also make other queries as follows:

$\text{Create}(\text{ID}_i)$: $\mathcal{S}$ needs to keep a table $L_U = (\text{ID}_i, x_i, X_i)$. After receiving the request from $\mathcal{A}$, $\mathcal{S}$ first checks if (ID_i, x_i, X_i) exists in L_U. If so, $\mathcal{S}$ returns X_i to A; otherwise, $\mathcal{S}$ checks whether the identity ID_i used by $\mathcal{A}$ is equal to the challenging identity ID_I, and if not, $\mathcal{S}$ generates x_i and X_i according to the proposed scheme; otherwise, $\mathcal{S}$ sets $X_i = Q$, stores $(\text{ID}_i, , X_i)$ into L_U, and returns X_i to $\mathcal{A}$

$\text{Sign}(\text{ID}_i, m_i)$: after receiving the request from $\mathcal{A}$, $\mathcal{S}$ first checks whether ID_i and ID_I are equal. If not, $\mathcal{S}$ generates the signature of the message m_i according to the proposed scheme; otherwise, $\mathcal{S}$ randomly chooses $s_i, k_i, \phi_i \in Z_q^*$, computes $C_i = (s_iG, m_iG + s_i(X_A + X_{CC}))$, $K_i = k_iG - \phi_i \cdot X_i$, sets $z_i = k_i$, stores $(\text{ID}_i, X_i, C_i, K_i, T_i, \phi_i)$ in L_h, and returns (K_i, z_i) to $\mathcal{A}$

$\text{Verify}(\text{ID}_i, m_i)$: $\mathcal{S}$ verifies the signature of the message m_i according to the proposed scheme

Finally, $\mathcal{A}$ can use $\mathcal{S}$ to forge a valid signature (C_i, K_i, z_i, T_i) of the message m_i with the identity ID_i. According to the forking lemma, $\mathcal{A}$ can obtain another valid signature (C_i, K_i, z_i', T_i), where $z_i = k_i + \phi_i \cdot x_i \bmod q$ and $z_i' = k_i + \phi_i' \cdot x_i \bmod q$. Therefore, we can obtain two equations:

$$\begin{aligned}
z_iG &= k_iG + \phi_ix_iG, \\
z_i'G &= k_iG + \phi_i'x_iG.
\end{aligned} \tag{27}$$

According to the above equations, we have $z_iG - z_i'G = (\phi_i - \phi_i')x_iG$, then $x_i = (z_i - z_i')(\phi_i - \phi_i')^{-1}$.

Finally, $\mathcal{S}$ gets the solution $x_i = (z_i - z_i')(\phi_i - \phi_i')^{-1}$ of the DL problem instance $(P, Q = x_iG)$.

To calculate the advantage of $\mathcal{S}$ solving the discrete logarithm problem, we define the following three events. (1) E_1: $\mathcal{S}$ does not terminate the game; (2) E_2: ID_i and ID_I are equal; (3) E_3: $\mathcal{A}$ outputs a valid signature.

Therefore, we have $\Pr[E_1] = (1 - (1/q_{H_4}))^{q_{\text{ext}}}$, $\Pr[E_2|E_1] = (1/q_{H_4})$, and $\Pr[E_3|E1 \wedge E_2] = \varepsilon$, where q_{H_4} and q_{ext} represent the number of H_4 query and Extract query, respectively. Therefore, the probability of $\mathcal{A}$ solving the discrete logarithm problem is

$$\begin{aligned}&\Pr\left[E1 \wedge E_2 \wedge E_3\right]\\&= \Pr\left[E_3 | E1 \wedge E_2\right] \cdot \Pr\left[E_2 | E_1\right] \cdot \Pr\left[E_1\right]\\&= \left(1 - \frac{1}{q_{H_4}}\right)^{q_{\text{ext}}} \cdot \frac{\varepsilon}{q_{H_4}}.\end{aligned} \tag{28}$$

Because ε cannot be ignored, the probability of $\mathscr{A}$ using $\mathcal{S}$ to solve discrete logarithm problem cannot be ignored. However, in the actual situation, the discrete logarithm problem is unable to solve; therefore, the conclusion cannot hold. As a result, our assumption does not hold. That is, the proposed scheme is secure against existential unforgeability under the adaptive chosen messages attacks (EUF-CMA).

6.2. Informal Security Analysis

(1) The proposed scheme provides anonymity for users. As wireless networks are increasingly used in the smart grid, communication channels may be open. It is easy for adversaries to intercept messages from communication channels. In the proposed authentication protocol, the identity of each smart meter is anonymous. Because the CDH problem is hard, the adversary cannot obtain a true identity without knowing the temporary random number. Therefore, the proposed scheme can protect the identity privacy of users.

(2) The proposed scheme ensures the confidentiality of the session key. In the proposed scheme, the session key in the proposed authentication protocol uses random numbers chosen by the smart meter and the aggregator. During each authentication phase, the smart meter and aggregator reselect new random numbers. Even if the adversary eavesdrops on the communication channel, it is difficult for the adversary to guess the session key or to calculate the session key from the messages transmitted over the network. Therefore, the proposed scheme ensures the confidentiality of the session key.

(3) The proposed scheme ensures the confidentiality of users' data. EC-ElGamal cryptosystem is used to encrypt the power consumption data. Assume that the DDH problem is hard, the EC-ElGamal cryptosystem is secure against IND-CPA. Therefore, an external eavesdropper cannot obtain any individual user's power consumption data. Furthermore, the adversary cannot infer the plaintext of the aggregated data in the aggregator's database and the control center's database. Therefore, the proposed scheme ensures the confidentiality of users' data.

(4) The proposed scheme ensures data integrity and authentication. The signature algorithm in the proposed scheme is provable secure. In practice, if an attacker wants to forge a signature, it would have to either crack the hash function or the discrete logarithm problem. In the proposed scheme, the signature algorithm uses a secure hash function and an elliptic curve, so that the possibility of both types of cracking is negligible. Therefore, the proposed scheme provides data integrity and authentication.

(5) The proposed scheme is secure under the attack of malware. Suppose an attacker successfully intercepts private information from the aggregator database by deploying malicious software in the aggregator system. Because the aggregator cannot completely decrypt the aggregated ciphertext, the attacker cannot obtain any single user's power consumption data. In addition, the attacker can also intercept private information from the control center. The decrypted plaintext of the control center is the sum of users' power consumption data, and the attacker cannot obtain the power consumption data of an individual user. Therefore, the proposed scheme can protect the user's power consumption data from malicious software.

(6) The proposed scheme can resist replay attacks. Because the messages whether in the authentication or data generation phase contain a timestamp, the aggregator can detect any replayed messages by verifying the validity of the timestamp. Therefore, the proposed scheme can resist replay attacks.

7. Performance Analysis

This section presents the performance comparison between the proposed scheme and other similar schemes in the data generation phase and data aggregation phase. Performance includes computation overhead and communication overhead. Experiments were all performed on a personal computer with Intel Core i5-7200U CPU @2.50 GHz, 12.00 GB memory, and Windows 10 operating system, based on the JPBC library.

7.1. Computation Cost. We compare the computation cost of the proposed scheme with that of Li et al. [27], Shen et al. [29], and Lang et al. [30]. For convenience, we define some notations and descriptions as shown in Table 2. Since CC is generally supposed to have enough computing power, we only compare the computation overhead of SM_i and AGG.

In Li's scheme [27], SM_i executes two EXP operations, one HTP operation, and one PMUL operation. Therefore, the runtime of SM_i is $2T_{\text{EXP}} + T_{\text{HTP}} + T_{\text{PMUL}}$. AGG executes one EXP operation, $(n+1)$PMUL operations, and one HTP operation. Therefore, the runtime of AGG is $T_{\text{EXP}} + (n+1)T_{\text{PMUL}} + T_{\text{HTP}}$.

In Shen's scheme [29], SM_i executes two EXP operations and one PMUL operation. Therefore, the runtime of SM_i is $2T_{\text{EXP}} + T_{\text{PMUL}}$. AGG executes $(n+1)$BP operations and nHTP operations. Therefore, the runtime of AGG is $(n+1)T_{\text{BP}} + nT_{\text{HTP}}$.

In Lang's scheme [30], SM_i executes $(l+1)$MUL – BGN operations and one PMUL operation. Therefore, the runtime of SM_i is $(l+1)T_{\text{MUL–BGN}} + T_{\text{PMUL}}$. AGG executes

Table 2: Notations and descriptions.

Notation	Description
EXP	The exponentiation operation in the Boneh-Goh-Nissim algorithm
MUL – BGN	The multiplication operation in the Boneh-Goh-Nissim algorithm
PMUL	The point multiplication operation in an elliptic curve group
PADD	The point addition operation in an elliptic curve group
HTP	The hash-to-point operation
BP	The bilinear pairing operation

Table 3: Computation costs at SM_i.

	User (ms)	AGG (ms)
Li's scheme [27]	$2T_{EXP} + T_{HTP} + T_{PMUL}$	$T_{EXP} + (n+1)T_{PMUL} + T_{HTP}$
Shen's scheme [29]	$2T_{EXP} + T_{PMUL}$	$(n+1)T_{BP} + nT_{HTP}$
Lang's scheme [30]	$(l+1)T_{MUL-BGN} + T_{PMUL}$	$(n+1)T_{PMUL}$
Our scheme	$(2l+2)T_{PMUL}$	$(n+1)T_{PMUL}$

$(n+1)$PMUL operations. Therefore, the runtime of AGG is $(n+1)T_{PMUL}$.

In our proposed scheme, SM_i executes $(2l+2)$PMUL operations. Therefore, the runtime of SM_i is $(2l+2)T_{PMUL}$. AGG executes $(n+1)T_{PMUL}$ operations. Therefore, the runtime of AGG is $(n+1)T_{PMUL} \approx (9.955n + 7.57)$ms.

Table 3 and Figure 5 show the computation costs comparisons among Li's scheme [27], Shen's scheme [29], Lang's scheme [30], and our proposed scheme.

7.2. Communication Cost. Because the size of q_1, q_2, q is 512 bits, 512 bits, and 160 bits, respectively, we can know that the size of $Z_n^*, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}, Z_q^*$ is 1024 bits, 1024 bits, 160 bits, 160 bits, and 160 bits. Assume that the timestamp and the identity are both 32 bits.

In Li's scheme [27], SM_i sends $(ID_i, CT_i, \delta_i, T)$ to AGG, where the length of CT_i is 1024 bits and the length of δ_i is 512 bits. Thus, the communication cost is $32 + 1024 + 512 + 32 = 1600$ bits.

In Shen's scheme [29], SM_i sends $(ID_i, ID_{AGG}, CT_i, \delta_i, T)$ to AGG, where the length of CT_i is 2048 bits and the length of δ_i is 160 bits. Thus, the communication cost is $32 + 32 + 2048 + 160 + 32 = 2304$ bits.

In Lang's scheme [30], SM_i sends $(ID_i, CT_i, \delta_i, T)$ to AGG, where the length of CT_i is 4096 bits and the length of δ_i is 512 bits when data has seven dimensions. Thus, the communication cost is $32 + 4096 + 512 + 32 = 4672$ bits.

In our proposed scheme, SM_i sends (ID_i, C_i, δ_i, T) to AGG, where the length of CT_i is 2240 bits and the length of δ_i is 512 bits when data has seven dimensions. Thus, the communication cost is $32 + 2240 + 512 + 32 = 2816$ bits.

As shown in Figures 5 and 6, the computation overhead in the aggregation phase of our proposed scheme has obvious advantages over Shen et al.'s [29] scheme. The communication cost is at the middle level compared with other schemes. Considering the security and reliability, it is reasonable to increase the communication cost. Therefore, the proposed scheme satisfies the requirement of security and performance for the smart grid.

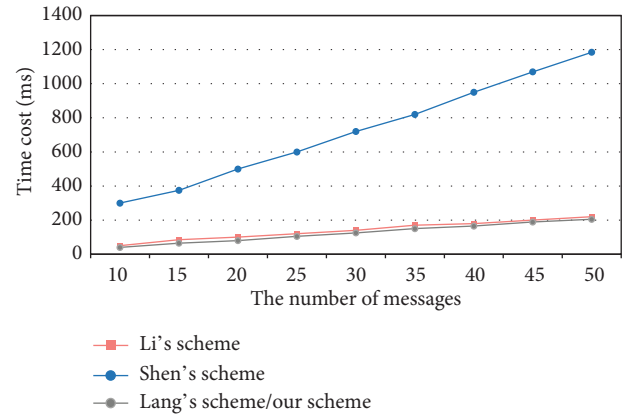

Figure 5: Computation costs at AGG.

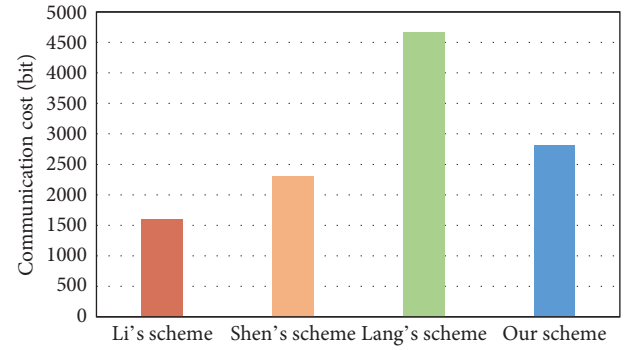

Figure 6: Communication costs.

8. Conclusion

In this paper, we propose a privacy-preserving multidimensional data aggregation scheme for a smart grid, which can aggregate multidimensional data and protect the user's

identity and data privacy. The analysis shows that the proposed scheme is provable secure and efficient. In addition, we will consider a more appropriate method of aggregating multidimensional data to improve the applicability of the proposed scheme in further work.

Data Availability

The data used to support the findings of this study are included within the article.

Conflicts of Interest

The authors have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

Our work was jointly supported by the National Natural Science Foundation of China (Nos. 61872051 and 61702067), the Chongqing Natural Science Foundation of China (No. cstc2020jcyj-msxmX0343), and the Venture & Innovation Support Program for Chongqing Overseas Returnees (No. CX2018122). There is no funding available.

References

[1] DOE US, "Grid 2030: a national vision for electricity's second 100 years," US DOE Report, pp. 137–140, DOE US, Washington, DC, USA, 2003.

[2] W. Meng, R. Ma, and H.-H. Chen, "Smart grid neighborhood area networks: a survey," *IEEE Network*, vol. 28, no. 1, pp. 24–32, 2014.

[3] R. Deng, Z. Yang, M.-Y. Chow, and J. Chen, "A survey on demand response in smart grids: mathematical models and approaches," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 570–582, 2015.

[4] 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.

[5] X. Tian, L. Li, C. Sun et al., "Review on privacy protection approaches in smart meter," *Journal of East China Normal University (Natural Science)*, vol. 2015, no. 5, pp. 46–60, 2015.

[6] D. Boneh, E. J. Goh, and K. Nissim, "Evaluating 2-DNF formulas on ciphertexts," in *Proceedings of the Theory of Cryptography Conference*, pp. 325–333, Springer, Cambridge, MA, USA, February 2005.

[7] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques*, pp. 223–238, Springer, Prague, Czech Republic, May 1999.

[8] N. Koblitz, "Elliptic curve cryptosystems," *Mathematics of Computation*, vol. 48, no. 177, p. 203, 1987.

[9] L. Chen, R. Lu, Z. Cao, K. AlHarbi, and X. Lin, "MuDA: multifunctional data aggregation in privacy-preserving smart grid communications," *Peer-to-Peer Networking and Applications*, vol. 8, no. 5, pp. 777–792, 2015.

[10] Q. Zhou, G. Yang, and L. He, "A secure-enhanced data aggregation based on ECC in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6701–6721, 2014.

[11] H. Bao and R. Lu, "A new differentially private data aggregation with fault tolerance for smart grid communications," *IEEE Internet of Things Journal*, vol. 2, no. 3, pp. 248–258, 2015.

[12] G. Shen, Y. Su, D. Zhang et al., "Secure and fine-grained electricity consumption aggregation scheme for smart grid," *TIIS*, vol. 12, no. 4, pp. 1553–1571, 2018.

[13] H. J. Jo, I. S. Kim, and D. H. Lee, "Efficient and privacy-preserving metering protocols for smart grid systems," *IEEE Transactions on Smart Grid*, vol. 7, no. 3, pp. 1732–1742, 2015.

[14] H. Li, X. Lin, H. Yang et al., "EPPDR: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 2053–2064, 2013.

[15] O. R. M. Boudia, S. M. Senouci, and M. Feham, "Elliptic curve-based secure multidimensional aggregation for smart grid communications," *IEEE Sensors Journal*, vol. 17, no. 23, pp. 7750–7757, 2017.

[16] S. Fu, J. Ma, H. Li, and Q. Jiang, "A robust and privacy-preserving aggregation scheme for secure smart grid communications in digital communities," *Security and Communication Networks*, vol. 9, no. 15, pp. 2779–2788, 2016.

[17] X. Liu, Y. Zhang, B. Wang, and H. Wang, "An anonymous data aggregation scheme for smart grid systems," *Security and Communication Networks*, vol. 7, no. 3, pp. 602–610, 2014.

[18] A. Abdallah and X. S. Shen, "A lightweight lattice-based homomorphic privacy-preserving data aggregation scheme for smart grid," *IEEE Transactions on Smart Grid*, vol. 9, no. 1, pp. 396–405, 2016.

[19] C. I. Fan, S. Y. Huang, and Y. L. Lai, "Privacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 1, pp. 666–675, 2013.

[20] H. Bao and R. Lu, "Comment on "rivacy-enhanced data aggregation scheme against internal attackers in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 12, no. 1, pp. 2–5, 2015.

[21] D. He, N. Kumar, S. Zeadally, A. Vinel, and L. T. Yang, "Efficient and privacy-preserving data aggregation scheme for smart grid against internal adversaries," *IEEE Transactions on Smart Grid*, vol. 8, no. 5, pp. 2411–2419, 2017.

[22] D. He, S. Zeadally, H. Wang, and Q. Liu, "Lightweight data aggregation scheme against internal attackers in smart grid using elliptic curve cryptography," *Wireless Communications and Mobile Computing*, vol. 2017, Article ID 3194845, 11 pages, 2017.

[23] E. Vahedi, M. Bayat, M. R. Pakravan, and M. R. Aref, "A secure ECC-based privacy preserving data aggregation scheme for smart grids," *Computer Networks*, vol. 129, pp. 28–36, 2017.

[24] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[25] Z. He, S. Pan, and D. Lin, "PMDA: privacy-preserving multi-functional data aggregation without TTP in smart grid," in *Proceedings of the 2018 17th IEEE International Conference on Trust, Security and Privacy in Computing and Communications*, pp. 1107–1114, IEEE, New York, NY, USA, August 2018.

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

[27] S. Li, K. Xue, Q. Yang et al., "PPMA: privacy-preserving multisubset data aggregation in smart grid," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 2, pp. 462–471, 2017.

[28] R. Lu, X. Liang, X. Li et al., "EPPA: an efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 9, pp. 1621–1631, 2012.

[29] 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.

[30] B. Lang, J. Wang, and Z. Cao, "Multidimensional data tight aggregation and fine-grained access control in smart grid," *Journal of Information Security and Applications*, vol. 40, pp. 156–165, 2018.