IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 June 2023, accepted 3 July 2023, date of publication 5 July 2023, date of current version 12 July 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3292586

RESEARCH ARTICLE

Fault-Tolerant Data Aggregation Scheme Supporting Fine-Grained Linear Operation in Smart Grid

ZICHAO SONG[1,2], TANPING ZHOU[1,2,3], WEIDONG ZHONG[1,2], DONG CHEN[1,2], LONGFEI LIU[1,2], AND XIAOYUAN YANG[1,2]

[1]College of Cryptography Engineering, Engineering University of PAP, Xi'an 710086, China
[2]Key Laboratory of PAP for Cryptology and Information Security, Xi'an 710086, China
[3]TCA Laboratory, State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences, Beijing 100080, China

Corresponding authors: Tanping Zhou (tanping2020@iscas.ac.cn) and Weidong Zhong (wdeast@163.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 62102452 and Grant 62172436, in part by the National Key Research and Development Program of China under Grant 2021YFB3100100, in part by the Innovative Research Team in Engineering University of PAP under Grant KYTD201805, and in part by the Natural Science Foundation of Shaanxi Province under Grant 2023-JC-YB-584.

ABSTRACT Smart grid is a combination of traditional power system engineering and information and communication technology. Smart grid provides users with convenient services through real-time data updates. Multi-dimensional data aggregation can be more flexible for statistical analysis of electricity information. However, most of the existing multi-dimensional data aggregation schemes require the participation of a trusted third party and do not support fault tolerance. In this paper, we propose a fault-tolerant data aggregation scheme supporting fine-grained linear operations in smart grid. Firstly, we used the Chinese remainder theorem to encode the user's multi-dimensional data and the corresponding weights. Secondly, we construct a privacy-preserving data aggregation scheme without a trusted third party, by combining paillier homomorphic encryption scheme and a secure key agreement protocol. Finally, we use the extended Shamir secret sharing scheme to construct a fault-tolerant data aggregation scheme that supports the reuse of shared key shares. Security analysis results show that our scheme satisfies semantic security and user data privacy protection. Experimental results show that compared with the existing multidimensional data aggregation schemes that require a trusted third party, our scheme does not increase additional computation and communication overhead.

INDEX TERMS Fault-tolerant, privacy protection, fine-grained linear operation, smart grid.

I. INTRODUCTION

Smart grid is a new generation of power grid that combines traditional power grid technology with information and communication technology to achieve efficient power generation, transmission, distribution, and control services [1], [2], [3]. Smart grid will play an increasingly important role in meeting user needs, improving data reliability, and ensuring energy control and management [4]. In recent years, smart grid has been developed rapidly in more and more countries. Smart grid is mainly through the real-time collection of user data to monitor and analyze the entire smart grid operation status. However, collecting user data can directly analyze user privacy behavior. For example, if a user uses zero electricity during a certain time of day, it can be inferred that no one is home during that time. If these data are obtained by some criminals, they can carry out illegal activities such as crimes at these time points. On the contrary, if power companies have access to the data, they can use big data technology to analyze it and improve the quality of service for customers. Obviously, exposing users' electricity consumption data will pose a threat to users' privacy [5]. Therefore, how to protect

The associate editor coordinating the review of this manuscript and approving it for publication was Salvatore Favuzza.

the private information of smart grid users from being leaked is particularly important [6].

Security and privacy are two important aspects of the smart grid. How to monitor regional power consumption without disclosing individual users' power consumption has become a research direction for scholars. Homomorphic encryption schemes ensure that algebraic operations on ciphertext are equivalent to direct operations on plaintext. Therefore, homomorphic encryption schemes have a wide range of applications in the privacy protection of smart grid data. Data aggregation uses the homomorphic properties of ciphertext in homomorphic encryption to aggregate multiple ciphertexts into one ciphertext to save bandwidth and reduce delay. Therefore, to solve the privacy protection problem in the smart grid, the idea of combining data aggregation and a homomorphic encryption algorithm is proposed.

With the development of science and technology and the Internet of Things, the computing efficiency of smart meters is getting higher and higher. The demand for user data from terminals is also becoming more and more detailed. A series of works are presented to protect the privacy of users in the smart grid. There are some very important properties in these privacy-preserving smart grid scenarios: no trusted third party, fine-grained linear operation, and fault tolerance. The early original scheme requires the participation of a trusted third party. However, in real life, it is difficult to find a trusted third party, and even if there is a trusted third party, the cost is expensive, so the data aggregation scheme without a trusted third party is particularly important. In literature [10], [12], [13], [14], [20], [21], and [31], the authors propose an aggregation scheme without a trusted third party by using a secure key agreement protocol, which reduces the running cost of the system; In literature [8], [14], [21], [22], and [31], the authors propose an aggregation scheme that allows users to flexibly join and leave the smart grid. The scheme reduces the operating costs for new households joining the smart grid; In literature [7], [8], [9], [14], [15], [20], [23], and [31], the authors propose a multidimensional data aggregation scheme. Multidimensional data (such as power consumption, time of use, variance, etc.) can provide a more detailed analysis basis for the terminal, so that the terminal can better develop electricity consumption strategy, etc.; In real life, there may be a failure of the user's smart meter. When the smart meter fails, the terminal hopes that the faulty smart meter will not affect the operation of the whole system. Therefore, in the literature [10], [21], [23], [24], and [28], the authors propose a fault-tolerant aggregation scheme. It solves the problem of correct decryption of the terminal when the smart meter is faulty. In the literature [15], the authors propose an aggregation scheme that supports linear homomorphic operations. The scheme enables the terminal to give different weights to different smart meters according to their needs for flexible statistical analysis.

The proposal of these schemes solves the different needs of the smart grid. However, none of these schemes support fine-grained linear homomorphic operations. With the rapid development of smart grids, the terminal's demand for statistical data is becoming more and more detailed. At the same time, the terminal has more and more functional requirements for the scheme. For example, a power company may want to calculate the total price of electricity that should be paid in a certain area, which requires a linear calculation of the electricity consumption information of the residents. Therefore, we propose a fault-tolerant data aggregation scheme that supports fine-grained linear operation. Our scheme solves the problem of the terminal achieving correct decrypting when the user's smart meter fails without a trusted third party. In addition, most cities have implemented step-by-step billing rules. Our scheme can calculate the electricity charge of interval electricity consumption, which is convenient for the power company to check the accounts, and prevent the internal personnel from tampering with the user ladder electricity consumption data and causing economic losses to the power company. As a result, our scheme is more practical.

Our contributions.

1). We apply the Chinese remainder theorem to encode the user's multidimensional data and corresponding multidimensional weights into one-dimensional data and one-dimensional weights. The linear operation of encoded one-dimensional data and one-dimensional weight is equivalent to the result of the linear operation of multidimensional data and corresponding weight. And the data of each dimension is independent of each other during terminal decryption.

2). Based on the paillier homomorphic encryption scheme and bilinear mapping, we construct a homomorphic data aggregation scheme that protects user privacy and supports batch verification. In addition, we use a secure key agreement protocol to construct a data aggregation scheme that does not require a trusted third party.

3). We use the extended Shamir secret sharing scheme to construct a data aggregation scheme with fault tolerance. When a user's smart meter fails, our scheme can achieve correct decryption. At the same time, our scheme does not reveal the secret shares of faulty users, which can prevent malicious attacks that intentionally compromise smart meters.

The remaining sections of this paper are arranged as follows: In section II, we present a number of existing works. In section III, we introduced the theoretical knowledge used in the article. In section IV, we introduced the system model, threat model, and design goal of our scheme. In section V, we present our scheme in detail. In section VI, we analyzed our scheme. In section VII, we conducted an experimental evaluation of the scheme. In section VIII, we draw our conclusions.

II. RELATED WORK

In this section, we introduce the smart grid in the related work of privacy protection schemes. In 2010, Li et al. [32] proposed a data aggregation method for privacy protection by combining an aggregation tree and a homomorphic encryption algorithm. The method with the minimum communication overhead ensures that all equipment is involved in polymerization, any equipment cannot get in the middle of the aggregate the results. In 2018, Lyu et al. [26] proposed a fog computing aggregation scheme, which uses fog nodes to collect and transmit data for efficient processing and calculation. This method greatly improves the efficiency of terminal computing in privacy-preserving data aggregation schemes. Therefore, in recent years, the method based on fog calculation has been widely used.

Multidimensional data contains multiple information about users, which can facilitate power companies to analyze the whole smart grid system in more detail and provide better services for users. At present, most schemes to achieve multi-dimensional data aggregation are mainly through the combination of super-increasing sequences and homomorphic encryption algorithm [7], [8], [30]. However, constructing multidimensional data using super-increasing sequences leads to an exponential increase in computational overhead and is less efficient. In the literature [25], [27], and [33], the author constructed a multidimensional data aggregation scheme by encrypting data of each dimension and generating different ciphertexts respectively. However, this method will cause a waste of corresponding resources in computation overhead and communication overhead. In literature [23], the author constructs a multidimensional data aggregation scheme through a special coding method. But this method gives each dimension a certain length. This coding method will cause a waste of space and is not convenient for flexible adjustment of the number of dimensions. In literature [14], the author used the Chinese remainder theorem to construct a multidimensional data aggregation scheme. The user's multidimensional data was encoded into one-dimensional data, which reduced the computational overhead during encryption and ensured that each dimension was independent of the other during decryption.

Compared with the above scheme, we propose a fault-tolerant data aggregation scheme that supports fine-grained linear operation. We use the Chinese Remainder theorem to encode the user's multidimensional data. Due to the linear homomorphism of the Chinese Remainder theorem. The terminal can successfully restore the result of each one-dimensional linear operation when decrypting. In addition, we use an extended Shamir secret sharing scheme to achieve fault tolerance. As shown in Table 1, we compare the existing scheme from six aspects: Data Confidentiality(DaC), No Trusted Authority(NTA), Dynamic Users(DyU), Multidimensional Data(MD), Linear Operation(LO), and Fault Tolerance(FT). As can be seen from Table 1, our scheme is more functional.

TABLE 1. Comparison of functions.

Scheme	DaC	NTA	DyU	MD	LO	FT
[7]	Yes	No	No	Yes	No	No
[8]	Yes	No	Yes	Yes	No	No
[9]	Yes	No	No	Yes	No	No
[10]	Yes	Yes	No	No	No	Yes
[11]	Yes	No	No	No	No	No
[12]	Yes	Yes	No	No	No	No
[13]	Yes	Yes	No	No	No	No
[14]	Yes	Yes	Yes	Yes	No	No
[15]	Yes	No	No	Yes	Yes	No
[20]	Yes	Yes	No	Yes	No	No
[21]	Yes	Yes	Yes	No	No	Yes
[22]	Yes	No	Yes	No	No	No
[23]	Yes	No	No	Yes	No	Yes
[24]	Yes	No	No	No	No	Yes
[28]	Yes	No	No	No	No	Yes
[31]	Yes	Yes	Yes	Yes	No	No
Our	Yes	Yes	Yes	Yes	Yes	Yes

III. PRELIMINARIES

Our scheme is based on the Chinese remainder theorem, paillier homomorphic encryption scheme, bilinear mapping, secret sharing scheme, and combinational mathematics. Next, we describe the preparatory knowledge used in our scheme.

A. CHINESE REMAINDER THEOREM

Let $b_1, b_2 \cdots b_n$ be the number of pairwise co-primes, $x_1, x_2 \cdots x_n$ and $y_1, y_2 \cdots y_n$ respectively n integers, then the following congruence equations has the following properties [17]:

$$\begin{cases} x \equiv x_1 \bmod b_1 \\ x \equiv x_2 \bmod b_2 \\ \vdots \\ x \equiv x_n \bmod b_n \end{cases} \tag{1}$$

$$\begin{cases} y \equiv y_1 \bmod b_1 \\ y \equiv y_2 \bmod b_2 \\ \vdots \\ y \equiv y_n \bmod b_n \end{cases} \tag{2}$$

The form of solution can be expressed as $x = x_1{B_1}'B_1 + x_2{B_2}'B_2 + \cdots + x_n{B_n}'B_n \bmod B$, $y = y_1{B_1}'B_1 + y_2{B_2}'B_2 + \cdots + y_n{B_n}'B_n \bmod B$, where $B = b_1b_2\cdots b_n$, $B_i = \frac{B}{b_i}$, ${B_i}' = {B_i}^{-1} \bmod b_i$, $1 \leq i \leq n$. Moreover, there is $x_i \cdot y_i = x \cdot y \bmod b_i, 1 \leq i \leq n$, and the linear operation of the corresponding dimension can be achieved by using the properties of the Chinese remainder theorem.

B. PAILLIER HOMOMORPHIC ENCRYPTION

Paillier public key encryption algorithm [18] is a popular homomorphic encryption that supports homomorphic addition.

– Key generation.Given security parameters κ, random generation of two large primes p and q, calculated $N = pq, \lambda = lcm(p-1, q-1)$. Defining functions

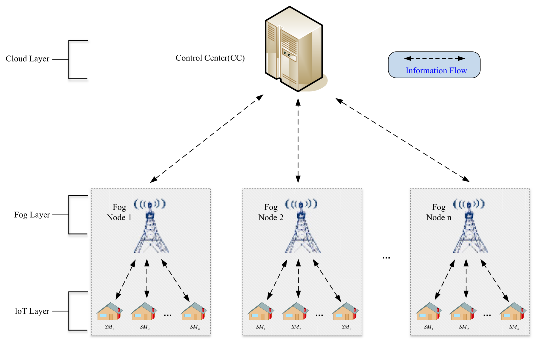


FIGURE 1. Smart grid system model.

$L(x) = \frac{x-1}{N}$, and randomly selecting the raw elements $g \in \mathbb{Z}^*_{N^2}$, make gcd $(L(g^\lambda mod\ N^2), N) = 1$ and calculate $\mu = (L(g^\lambda \bmod N^2))^{-1} \bmod N$. Then, the public key of the encryption algorithm is $PK = (N, g)$, the private key is $SK = (\lambda, \mu)$.

–Encryption.For any plaintext$m \in {}_N$, select a random integer r, where $0 < r < N, r \in \mathbb{Z}^*_{N^2}$. That is, r has a multiplication inverse in the remainder of N^2. Calculate ciphertext $c = g^m r^N \bmod N^2$.

–Decryption.For a given ciphertext $c \in \mathbb{Z}^*_{N^2}$, calculate the plaintext$m = L(c^\lambda \bmod N^2) \cdot \mu \bmod N$.

–Homomorphic addition.For any plaintext $m_1, m_2 \in \mathbb{Z}_N$, integers r_1, r_2 are randomly selected to satisfy $0 < r_1, r_2 < N, r_1, r_2 \in \mathbb{Z}^*_{N^2}$, which can be obtained after encryption $c_1 = g^{m_1} {r_1}^N \bmod N^2$,$c_2 = g^{m_2} {r_2}^N \bmod N^2$. Therefore, there is $c_1 \cdot c_2 = g^{m_1+m_2}(r_1 \cdot r_2)^N \bmod N^2$, $c_1 \cdot c_2$ can be decrypted as $D[c_1 \cdot c_2] = m_1 + m_2 \bmod N$. That is, the result after ciphertext multiplication and decryption is equal to the sum of the plaintext. Therefore, paillier homomorphic encryption algorithm is mainly used in the scenario of plaintext accumulation with privacy protection.

C. BILINEAR PAIRING

Let G_1, G_2 be a cyclic group that satisfies the order of a large prime number P, in which a pairing relation $e : G_1 \times G_1 \to G_2$ is defined to meet the following conditions [19]:

-Bilinear.For any $g, h \in G_1, a, b \in \mathbb{Z}_p$, there is a $e(g^a, h^b) = e(g, h)^{ab}$.

–Computability. For any $g, h \in G_1, a, b \in \mathbb{Z}_p$, there is an efficient polynomial time algorithm to calculate the value of $e(g^a, h^b), e(g, h)^{ab}$.

–Non-Degeneracy. For any $g, h \in G_1$, there is a $e(g, h) \neq 1_{G_2}$.

D. SECRET SHARING SCHEME

In this section, we present the Shamir secret sharing scheme and its extension.

1) SHAMIR SECRET SHARING SCHEME

Shamir secret sharing scheme [29] is a threshold secret sharing scheme based on the Lagrange interpolation theorem, which mainly includes secret distribution and secret reconstruction.

a: SECRET DISTRIBUTION STAGE

Let $s \in \mathbb{Z}_p$ (where p is a large prime integer) be the secret information to be shared, and the distributor randomly selects $k - 1$ coefficients $a_i \in \mathbb{Z}_p (1 \leqslant i \leqslant k - 1)$. Let $a_0 = s$, the distributor construct a polynomial of the degree $k - 1$ $f(x) = \sum_{i=0}^{k-1} a_i \cdot x^i \bmod p$. Then, the distributor computer $y_i = f(x_i), (1 \leqslant i \leqslant n)$, where x_i is the ID information of the users to be distributed and n is the total number of users to be shared. Finally, the distributor sends y_i to the corresponding user as a shared secret share.

b: SECRET RECONSTRUCTION STAGE

The reconstructor collects y_i of k users, and the secret information s can be restored using Lagrange interpolation $s = f(0) = \sum_{i=1}^{k} L_i \cdot y_i \bmod p$,where $L_i = \prod_{j=1, j\neq i}^{k} \frac{-x_j}{x_i - x_j} \bmod p$. Shamir secret sharing scheme can not restore polynomial $f(x)$ for any secret share of less than k sharing users, so it can not obtain secret information $s = f(0)$. Therefore, Shamir secret sharing scheme is resistant to conspiratorial attacks with fewer than k sharing users.

2) EXTENDED SHAMIR SECRET SHARING SCHEME

When using Shamir secret sharing scheme for secret refactoring, once the shared user provides the secret share, the shared user's secret share will be disclosed. Therefore, Wu et al. [24] proposed an extended Shamir secret sharing scheme that supported secret share reuse based on the Shamir secret sharing scheme.

a: SECRET DISTRIBUTION STAGE

First, the distributor randomly selects a large prime number p and two large prime factors u, v of $p - 1$, and calculates $N = uv$. In the finite field $GF(p)$, the generator g of order N is selected. Then, the distributor shares the secret information s in the module N and sends y_i to the relevant participant (same as Shamir secret sharing scheme). Then, if the distributor wants to share a new secret message s' at the time T_s, the distributor calculates $\Delta_{T_s} = s' - g^{r \cdot s} \bmod p$, where $r = H(T_s)$ is the time-dependent blind factor and H is the hash function. Finally, the distributor publishes Δ_{T_s}.

b: SECRET RECONSTRUCTION STAGE

When the reconstructor needs to reconstruct secret information s', it collects the secret share of k users. In this case, the sharing users do not provide the shared secret share y_i, but provide the equivalent blind share $Y_i = g^{r \cdot y_i} \bmod p$. Reconstructor computation

$$s' = \prod_{i=1}^{k} {Y_i}^{L_i} + \Delta_{T_s} \bmod p$$

$$\begin{aligned} &= \prod_{i=1}^{k} (g^{r \cdot y_i})^{L_i} + \Delta_{T_s} \bmod p \\ &= g^{r \cdot \sum_{i=1}^{k} y_i \cdot L_i} + \Delta_{T_s} \bmod p \\ &= g^{rs} + \Delta_{T_s} \bmod p \\ &= s' \bmod p, \end{aligned} \tag{3}$$

where $k \in \mathbb{Z}, g^N = 1 \bmod p$. Thus, secret information s' can be reconstructed without disclosing the share.

IV. SYSTEM DESIGN

In this section, we introduce the structure of smart grid system from three aspects: system model, threat model, and design goals.

A. SYSTEM MODEL

In our scheme, there are three main participants: smart meters (SM), fog nodes (FN), and control center (CC). As shown in Figure 1, the information transmission process of the three participants is given. The roles and functions of the three parties in the system are as follows.

–Smart meter(SM). Smart meters are smart devices that power companies install in customers' homes. It is responsible for collecting and encrypting the user's electricity data, and then transmitting the encrypted data to the nearest FN.

-Fog node(FN). FN has powerful storage and computing power. It collects, processes, and aggregates data from smart meters. Then, FN transmits the aggregated data to CC.

–Control center(CC). CC is responsible for receiving the data from FN, verifying and decrypting it. Then, CC processes this data. By analyzing the decrypted results, CC can know the operation status of the whole smart grid in real-time, check whether the system is faulty, adjust the operation strategy, etc.

B. THREAT MODEL

In our scheme, all parties are semi-honest. Typically, participants will process the data exactly as the protocol requires. At the same time, participants will try to obtain sensitive information about users. We assume that the attacker has the following capabilities.

–An attacker can intercept the communication data among SM, FN, and CC and try to obtain the user's sensitive information from these data.

–An attacker can invade FN's and CC's databases, to steal the user's data and related parameters. The attacker attempts to recover the user's sensitive information from the data.

–In order to obtain the sensitive information of a specific user, the attacker can cooperate with some users to conduct a collusive attack. In addition, the attacker can also forge the relevant user identity, inject false information into the system and destroy the integrity of the system.

C. DESIGN GOALS

Our goal is to design a privacy-preserving data aggregation scheme that supports linear operations on multidimensional data. Specifically, our scheme aims to achieve the following functions.

– Security. User data should be safe in the whole system transmission process. Specifically, it includes user data confidentiality, integrity, and terminal accessibility. External attackers cannot tamper with or forge user data and the system can check the validity of user identities.

–Privacy. The power of the user data is sensitive. The electricity consumption data of individual users cannot be obtained by any participant. Prevent the leakage of users' privacy information through electricity data.

–Practicality. The designed scheme can realize the linear operation of user multidimensional data. No trusted third party involvement is required in the entire process. In addition, considering the limited computing power of smart meters, the designed encryption algorithm should be efficient.

–Fault tolerance. In actual situations, some smart meters may fail, leading to incorrect decryption, so the scheme should have a certain degree of fault tolerance.

V. OUR SCHEME

In this section, we introduce our scheme in detail. As shown in Figure 2, shows the flow chart of our scheme. Table 2 lists our scheme using acronyms and symbols and their meanings.

A. SCHEME CONSTRUCTION

Our goal is to design a privacy-preserving data aggregation scheme that supports linear operations on multidimensional data. Specifically, our scheme aims to achieve the following functions.

1) INITIALIZATION STAGE

Step 1: Given the security parameter k, the system randomly selects two large prime numbers p, q. The system user set $\mathbb{U} = \{u_1, u_2, \cdots, u_n\}$ generates related parameters $(G_1, G_2, e, h, \omega_i)$. Where the group G_1, G_2 satisfies the bilinear relation $e : G_1 \times G_1 \rightarrow G_2$ and h is a generator of the group G_1.

Step 2: CC calculates $N = p \cdot q, \lambda = lcm(p-1, q-1)$, select $g = N+1$, large prime $Q = \varepsilon N+1$, where ε is a small integer. CC chooses l coprime large prime $B = \{b_1, b_2, b_3 \cdots b_l\}$ such that $b_j \geqslant nd_{\max}\omega_{\max}$, $0 \leqslant j \leqslant l$, where $d_{\max}$ is the upper bound of the user's single dimension data, $\omega_{\max}$ is the upper bound of the user's single dimension weight, n is the number of users, and computes $\widehat{B} = b_1 b_2 \cdots b_l$, $B_j = \frac{\widehat{B}}{b_j}$,$B_j' = B_j^{-1} \bmod b_j$,$0 \leqslant j \leqslant l$. CC randomly selects n blind factors and assigns them to these users, hash function $H_1 : \{0, 1\}^* \rightarrow G_1, H_2 : \{0, 1\}^* \rightarrow Z_N{}^*$. Meanwhile, CC assigns weight ω_i to different users according to needs, ω_i satisfy $\omega_i = \omega_{i1} B_1 B_1' + \omega_{i2} B_2 B_2' + \omega_{i3} B_3 B_3' + \cdots + \omega_{il} B_l B_l' \bmod \widehat{B}$.

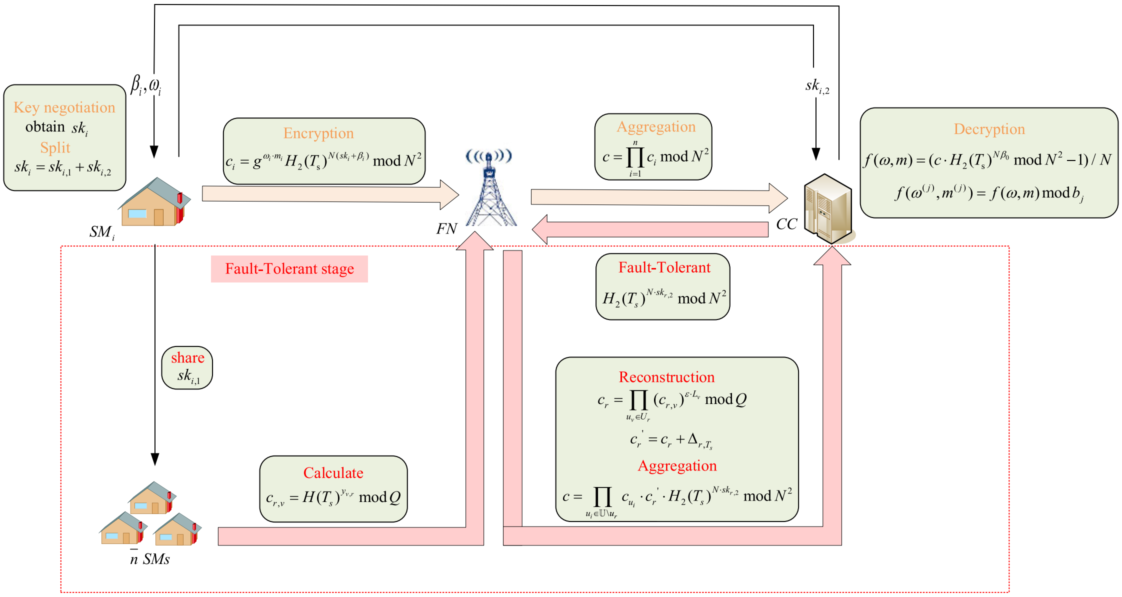

Key negotiation
obtain sk_i
Split
$sk_i = sk_{i,1} + sk_{i,2}$
β_i, ω_i
SM_i
Encryption
$c_i = g^{\omega_i \cdot m_i} H_2(T_s)^{N(sk_i+\beta_i)} \bmod N^2$
FN
Aggregation
$c = \prod_{i=1}^{n} c_i \bmod N^2$
$sk_{i,2}$
CC
Decryption
$f(\omega, m) = (c \cdot H_2(T_s)^{N\beta_0} \bmod N^2 - 1) / N$
$f(\omega^{(j)}, m^{(j)}) = f(\omega, m) \bmod b_j$
Fault-Tolerant stage
Fault-Tolerant
$H_2(T_s)^{N \cdot sk_{r,2}} \bmod N^2$
share
$sk_{i,1}$
Reconstruction
$c_r = \prod_{u_v \in U_r} (c_{r,v})^{\varepsilon \cdot L_v} \bmod Q$
$c_r' = c_r + \Delta_{r,T_s}$
Aggregation
$c = \prod_{u_i \in \mathbb{U} \setminus u_r} c_{u_i} \cdot c_r' \cdot H_2(T_s)^{N \cdot sk_{r,2}} \bmod N^2$
Calculate
$c_{r,v} = H(T_s)^{y_{v,r}} \bmod Q$
$\overline{n}$ SMs

FIGURE 2. Flow chart of our scheme.

TABLE 2. Notations in our scheme.

Notation	Description
SM_i	Smart Meters
FN	Fog Node
CC	Control Center
l	Dimension of data
m_{il}	l-dimensional data
m_i	$m_i \xleftarrow{CRT} (m_{i1}, m_{i2}, \cdots, m_{il})$
$\omega_{i,l}$	The weight corresponding to the l-dimension data
ω_i	$\omega_i \xleftarrow{CRT} (\omega_{i,1}, \omega_{i,2}, \omega_{i,3}, \cdots, \omega_{i,l})$
x_i	User signature key
Y_i	User authenticated key
ID_i, ID_f	User and FN identity
c	Ciphertext after FN aggregation
$f(\omega, m)$	The result of linear aggregation of all user data and weights
$f(\omega^{(j)}, m^{(j)})$	The aggregate result of all users' j-dimension data and corresponding weight linear operation
sk_i	Encryption key
$sk_{i,1}, sk_{i,2}$	Split key,$sk_i = sk_{i,1} + sk_{i,2}$
Δ_{i,T_s}	The parameters computed after the users share $sk_{i,1}$
$y_{*,i}$	The corresponding user shares the secret share of the private key $sk_{i,1}$ of the first part of the user u_i
β_i	User blind factor
$\mathbb{k}_i$	User's key set
T, T_s	Current time and time slot
H_1, H_2	$H_1 : \{0,1\}^* \to G_1, H_2 : \{0,1\}^* \to Z_N{}^*$
σ_i, σ_f	Signature value of the user and FN
λ	CC's private key
b_i	Large prime selected by Chinese remainder theorem

Be denoted as $\omega_i \xleftarrow{CRT} (\omega_{i,1}, \omega_{i,2}, \omega_{i,3}, \cdots, \omega_{i,l})$, where l is the number of dimensions and $\omega_{i,l}$ is the weight of data in dimension l. Then the CC discloses the relevant parameters $\{l, N, Q, g, G_1, G_2, h, e, H_1, H_2, \varepsilon, \widehat{B}, B_j, B_j'\}$ and the CC retains the private key λ.

Step 3: Each user $u_i \in \mathbb{U}$ registers randomly choosing $x_i \in_R Z_N{}^*$ as its private key and computing $Y_i = h^{x_i}$ as its public key, and then FN registers, randomly choosing $x_f \in_R Z_N{}^*$ as its private key and computing $Y_f = h^{x_f}$ as its public key.

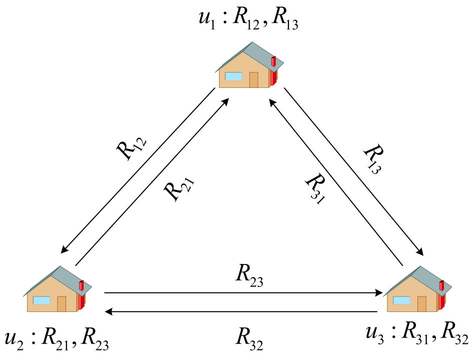
Step4: User $u_i \in \mathbb{U}$ randomly selects $R_{ij} \in_R Z_{N^2}{}^*$, where $1 \leq j \leq n-1$. $\{R_{i1}, \cdots, R_{i,n-1}\}$ form the user's shared key set. Through the secure channel, users send R_{ij} to user j. User $u_i \in \mathbb{U}$ receives the shared key of other users, which forms the key set of users $\{R_{1i}, \cdots, R_{ni}\}$. The user's key set $\left\{R_{i1}, \cdots, R_{i,n-1}, R_{1i}, \cdots, R_{ni}\right\}$ is recorded $\mathbb{k}_i$, and this user $u_i \in \mathbb{U}$ uses his key set to calculate the encrypted key $sk_i = R_{i1} + \cdots + R_{i,n-1} - R_{1i} - \cdots - R_{ni}$, clearly $\sum_{i=1}^{n} sk_i = 0$. To illustrate this, three user examples are given, as shown in Figure 3.

Step 5: The user $u_i \in \mathbb{U}$ splits his private key sk_i into two parts $sk_{i,1}$ and $sk_{i,2}$, satisfying $sk_i = sk_{i,1} + sk_{i,2}$. The user u_i randomly selects $\overline{n}(\overline{n} \leqslant n)$ users in this area and shares $sk_{i,1}$ with $\overline{n}$ users by using the Shamir secret sharing scheme extended under the module N. The share of each user is denoted as $y_{*,i}$ ($*$represents the general user symbol, and $y_{*,i}$ represents the secret share of the first part $sk_{i,1}$ of the user u_i's private key reserved by the corresponding user). At the same time, u_i sends $sk_{i,2}$ to CC using the secure channel. At time interval T_s, the user u_i calculates $\Delta_{i,T_s} = H_2(T_s)^{N \cdot sk_{i,1}} \bmod N^2 - H_2(T_s)^{\varepsilon \cdot sk_{i,1}} \bmod Q$ and exposes Δ_{i,T_s}.

In addition, two linear functions are defined.

$$f(\omega, m) = \sum_{i=1}^{n} \omega_i \cdot m_i \tag{4}$$

$$f(\omega^{(j)}, m^{(j)}) = \sum_{i=1}^{n} \omega_{ij} \cdot m_{ij}, 1 \leqslant j \leqslant l \tag{5}$$



u_1 computes secret key : $sk_1 = R_{12} + R_{13} - R_{21} - R_{31}$

u_2 computes secret key : $sk_2 = R_{21} + R_{23} - R_{12} - R_{32}$

u_3 computes secret key : $sk_3 = R_{31} + R_{32} - R_{13} - R_{23}$

therefore $sk_1 + sk_2 + sk_3 = 0$

FIGURE 3. The process of generating the user key.

2) REPORTING PHASE

When the user $u_i \in \mathbb{U}$ needs to send smart meter data to FN at the time interval T_s, the user collects multidimensional data $m_i \xleftarrow{CRT} (m_{i1}, m_{i2}, \cdots, m_{il})$ at this time, where m_{il} is the data of the user u_i corresponding to the l dimension, m_i is the result of transformation and calculation through the Chinese remainder theorem, m_i satisfies $m_i = m_{i1}B_1B_1{}' + m_{i2}B_2B_2{}' + m_{i3}B_3B_3{}' + \cdots + m_{il}B_lB_l{}' \bmod \widehat{B}$.Then, the user encrypts the message m_i by $c_i = g^{\omega_i \cdot m_i} H_2(T_\text{s})^{N(sk_i+\beta_i)} \bmod N^2$ and generates the signature value $\sigma_i = H_1(ID_i \parallel T \parallel c_i)^{x_i}$ with his private key x_i, which T is the current time, which can be used to defend against replay attacks. After that, the user sends the data (c_i, σ_i, T, ID_i) to the nearest FN.

3) READING PHASE

Firstly, FN verifies the legitimacy and integrity of the message received from the users. FN verifies the following formula $e(\sigma_i, h) \overset{?}{=} e(H_1(ID_i \parallel T \parallel c_i), Y_i)$. To improve the efficiency of verification, FN can verify the legitimacy and integrity of the current user set in batch. Verify that the following formula

$$\begin{aligned} e(\sum_{i=1}^{n} \sigma_i, h) &= e(\sum_{i=1}^{n} H_1(ID_i \parallel T \parallel c_i)^{x_i}, h) \\ &= \sum_{i=1}^{n} e(H_1(ID_i \parallel T \parallel c_i)^{x_i}, h) \\ &= \sum_{i=1}^{n} e(H_1(ID_i \parallel T \parallel c_i), Y_i) \end{aligned} \quad (6)$$

If the verification fails, FN asks the user to resend the data. Secondly, after the batch verification is passed, FN aggregates the data. Due to $\sum_{i=1}^{n} sk_i = 0$, we have

$$\begin{aligned} c &= \prod_{i=1}^{n} c_i \bmod N^2 \\ &= g^{\sum_{i=1}^{n} \omega_i \cdot m_i} H_2(T_\text{s})^{\sum_{i=1}^{n} N(sk_i+\beta_i)} \bmod N^2 \\ &= g^{f(\omega,m)} H_2(T_\text{s})^{\sum_{i=1}^{n} N\beta_i} \bmod N^2, \end{aligned} \quad (7)$$

where $f(\omega, m) = \sum_{i=1}^{n} \omega_i \cdot m_i$ represents the value of all users' message aggregations. Then, FN signs the aggregated value with its private key and generates the signature value $\sigma_f = H_1(ID_\text{f} \parallel T \parallel c)^{x_f}$. Finally, FN sends (c, σ_f, T, ID_f) to CC.

4) DECRYPTION PHASE

After receiving the report from FN, CC first verifies the formula $e(\sigma_f, h) \overset{?}{=} e(H_1(ID_f \parallel T \parallel c), Y_f)$. Then, CC calculates that β_0 satisfies $\sum_{i=1}^{n} \beta_i + \beta_0 = 0 \bmod \lambda$ to decrypt the cipher c using its private key λ.

$$\begin{aligned} f(\omega, m) &= (c \cdot H_2(T_\text{s})^{N\beta_0} \bmod N^2 - 1)/N \\ &= (g^{f(\omega,m)} H_2(T_\text{s})^{\sum_{i=1}^{n} N\beta_i} \cdot H_2(T_\text{s})^{N\beta_0} \bmod N^2 - 1)/N \\ &= (g^{f(\omega,m)} \bmod N^2 - 1)/N \\ &= ((N+1)^{f(\omega,m)} \bmod N^2 - 1)/N \\ &= (1 + f(\omega, m)N - 1)/N \\ &= f(\omega, m) \end{aligned} \quad (8)$$

Then, CC computes

$$\begin{aligned} & f(\omega^{(j)}, m^{(j)}) \\ &= f(\omega, m) \bmod b_j \\ &= \sum_{i=1}^{n} \omega_i \cdot m_i \bmod b_j \\ &= \sum_{i=1}^{n} [(\sum_{j=1}^{l} \omega_{ij} \cdot B_jB_j{}') \cdot (\sum_{j=1}^{l} m_{ij} \cdot B_jB_j{}') \bmod \widehat{B}] \bmod b_j \\ &= \sum_{i=1}^{n} \omega_{ij} \cdot m_{ij}, \end{aligned} \quad (9)$$

where $1 \leqslant j \leqslant l$. The sum of linear operations of all users corresponding to each dimension data can be obtained, which provides support for CC to conduct dynamic analysis and adjust power supply strategy.

B. FAULT TOLERANCE

1) THEORETICAL ANALYSIS

To study the probability of smart grid failure and successful recovery. We assume that the smart meters are all independent of each other and the failure rate of the smart meters is θ. As long as there are at least k meters of all the $\overline{n}$ meters works, we can recover the equivalent ciphertext by Shamir secret sharing scheme. Therefore, the probability of successfully recovering the equivalent ciphertext is

$$P=\sum_{i=k}^{\overline{n}} C_{\overline{n}}^{i}\cdot\theta^{\overline{n}-i}\cdot(1-\theta)^{i} \tag{10}$$

Furthermore, we present two examples to illustrate the probability of successful recovery. When $\theta=3\%$, $k=3$, $\overline{n}=5$, we have $P=99.97\%$. When $\theta=5\%$, $k=13$, $\overline{n}=20$, we have $P=99.99\%$.

2) DETAILS OF FAULT TOLERANCE

At the time T_{s}, FN could not collect users' information due to the damage to electric meters for some users, leading to incorrect decryption. Our scheme was fault-tolerant, and we used the extended Shamir secret sharing scheme to achieve correct decryption.

In order to illustrate the fault tolerance of our scheme, assume that at the time T_{s}, the user $u_r\in\mathbb{U}$ fails and cannot send his information to FN. To complete decryption, FN collects the secret share that the user u_r shares with other users. After the user $u_v(v=1,2,3,\cdots,\overline{n})$ receives the request, calculate $c_{r,v}=H(T_s)^{y_{v,r}} \bmod Q$ ($y_{v,r}$ means useru_v shares user u_r's secret share), and collect at least k effective secret shares about the user u_r from the user u_v, denoted as U_r. FN calculates

$$L_v=\prod_{u_w\in U_r,u_w\neq u_v}\frac{-u_w}{u_v-u_w} \bmod N \tag{11}$$

$$c_r=\prod_{u_v\in U_r}(c_{r,v})^{\varepsilon\cdot L_v} \bmod Q \tag{12}$$

At the same time, the parameter Δ_{r,T_s} of the user u_r is used to calculate $c_r{}'=c_r+\Delta_{r,T_s}$, while CC computes the second part share $H_2(T_s)^{N\cdot sk_{r,2}} \bmod N^2$ and sends it to FN for aggregation calculation

$$c=\prod_{u_i\in\mathbb{U}\backslash u_r}c_{u_i}\cdot c_r{}'\cdot H_2(T_s)^{N\cdot sk_{r,2}} \bmod N^2 \tag{13}$$

3) CORRECTNESS

The correctness of fault tolerance:

$$\begin{aligned}c&=\prod_{u_i\in\mathbb{U}\backslash u_r}c_{u_i}\cdot c_r{}'\cdot H_2(T_s)^{N\cdot sk_{r,2}} \bmod N^2\\&=\prod_{u_i\in\mathbb{U}\backslash u_r}c_{u_i}\cdot\Big(\prod_{u_v\in U_r}(c_{r,v})^{\varepsilon\cdot L_v} \bmod Q+\Delta_{r,T_s}\Big)\\&\quad\cdot H_2(T_s)^{N\cdot sk_{r,2}} \bmod N^2\\&=\prod_{u_i\in\mathbb{U}\backslash u_r}c_{u_i}\cdot\Big(H_2(T_s)^{\varepsilon\sum_{u_v\in U_r}y_{v,r}\cdot\prod_{u_w\in U_r,u_w\neq u_v}\frac{-u_w}{u_v-u_w} \bmod N} \bmod Q\\&\quad+\Delta_{r,T_s}\Big)\cdot H_2(T_s)^{N\cdot sk_{r,2}} \bmod N^2\\&=\prod_{u_i\in\mathbb{U}\backslash u_r}c_{u_i}\cdot\big(H_2(T_s)^{\varepsilon(sk_{r,1}+k'N)} \bmod Q+\Delta_{r,T_s}\big)\\&\quad\cdot H_2(T_s)^{N\cdot sk_{r,2}} \bmod N^2\end{aligned} \tag{14}$$

where $k'\in\mathbb{Z}$. Because of $H_2(T_s)^{\varepsilon N} \bmod Q=1$, we have $H_2(T_s)^{\varepsilon k'N} \bmod Q=1$. Further have

$$\begin{aligned}c&=\prod_{u_i\in\mathbb{U}\backslash u_r}c_{u_i}\cdot(H_2(T_s)^{N\cdot sk_{r,1}})\cdot H_2(T_s)^{N\cdot sk_{r,2}} \bmod N^2\\&=\prod_{u_i\in\mathbb{U}\backslash u_r}c_{u_i}\cdot H_2(T_s)^{N\cdot sk_r} \bmod N^2\\&=g^{\sum_{u_i\in\mathbb{U}\backslash u_r}\omega_{u_i}\cdot m_{u_i}}H_2(T_{\mathrm{s}})^{\sum_{u_i\in\mathbb{U}\backslash u_r}N\cdot\beta_{u_i}} \bmod N^2\end{aligned} \tag{15}$$

CC updates β_0 to satisfy $\sum_{u_i\in\mathbb{U}\backslash u_r}\beta_{u_i}+\beta_0=0 \bmod \lambda$ and decrypts to obtain

$$\begin{aligned}f(\omega_{u_i},m_{u_i})&=(c\cdot H_2(T_{\mathrm{s}})^{\beta_0} \bmod N^2-1)/N\\&=(g^{\sum_{u_i\in\mathbb{U}\backslash u_r}\omega_{u_i}\cdot m_{u_i}} \bmod N^2-1)/N\\&=(1+N\sum_{u_i\in\mathbb{U}\backslash u_r}\omega_{u_i}\cdot m_{u_i}-1)/N\\&=\sum_{u_i\in\mathbb{U}\backslash u_r}\omega_{u_i}\cdot m_{u_i}\end{aligned} \tag{16}$$

where $u_i\in\mathbb{U}\backslash u_r$. By $f(\omega_{u_i},m_{u_i})$ module $b_j(1\leqslant j\leqslant l)$, $f(\omega_{u_i}{}^{(j)},m_{u_i}{}^{(j)})$ can be obtained

$$\begin{aligned}f(\omega_{u_i}{}^{(j)},m_{u_i}{}^{(j)})&=f(\omega_{u_i},m_{u_i}) \bmod b_j\\&=\sum_{u_i\in\mathbb{U}\backslash u_r}\omega_{u_i}\cdot m_{u_i} \bmod b_j\\&=\sum_{u_i\in\mathbb{U}\backslash u_r}\omega_{u_i,j}\cdot m_{u_i,j},\end{aligned} \tag{17}$$

where $(1\leqslant j\leqslant l)$. End proof.

C. DYNAMIC USER MANAGEMENT

Our scheme supports dynamic user join and revoke. User failure can be considered as user revocation.

1) USER REVOKE

When the user $u_{n'}\in\mathbb{U}$ revokes, FN broadcasts to revoke the user $ID_{n'}$. Each user removes the shared key $R_{n'i}$ of the revocation user and the shared key $R_{jn'}$ sent to the user in its own key set. Then, others re-update their set of shared keys, calculating the private key at this point in encryption. CC calculates the $\sum_{i=1}^{n}\beta_i+\beta_0{}'-\beta_{n'}=0 \bmod \lambda$ update decryption blind factor $\beta_0{}'$.

2) USER JOIN

When the user $u_{i'}$ joins, the FN broadcast adds user identity information $ID_{i'}$. Then, other users update their key sets as initialized. CC randomly generates a blind factor $\beta_{i'}\in_R Z_N{}^*$, sends it to the user, and updates CC's decryption blind factor $\beta_0{}'$ according to $\sum_{i=1}^{n}\beta_i+\beta_0{}'+\beta_{i'}=0 \bmod \lambda$. At the same time, CC assigns weight $\omega_{i'}$ to the user $u_{i'}$.

D. BASED ON THE SCHEME OF THE APPLICATION

The step electricity price is the step increasing electricity price, which means that the average household electricity consumption is set into several steps or grades of pricing calculation costs. After the oil crisis in the 1970s, Japan, South Korea, and some parts of the United States adopted the step pricing system for residential electricity. The less electricity used, the lower the price, while the more electricity used, the higher the price. In real life, almost all regions have implemented the classification of the ladder pricing method. Step electricity consumption is generally divided into three types: industrial and commercial electricity consumption, agricultural electricity consumption, and residential electricity consumption. Each type carries out different step charging standards. Implementing step-increasing prices for residential electricity consumption can improve energy efficiency. Segmented electricity can realize differentiated pricing of market segments and improve electricity efficiency. The establishment of a tiered pricing mechanism of "multiple users pay more" will help to form a social consensus on energy conservation and emission reduction and promote the construction of a resource-conserving and environment-friendly society. Therefore, the real-time statistics of electricity and electricity consumption in the region have a certain significance for the company to conduct dynamic analysis and adjust the electricity price. At the same time, collecting users' electricity bills in real time can prevent internal staff from tampering with the power consumption of different types of users at different levels of the company.

In order to better illustrate the practicability of our scheme, as shown in Figure 4, we briefly illustrate the step accounting process with specific examples. It is assumed that there are three types of electricity consumption in a region: industrial and commercial electricity consumption, agricultural electricity consumption, and residential electricity consumption, and each type of electricity consumption executes different charging standards. Using our linear homomorphic operation scheme, the electricity consumption of all users of different types in the local region can be counted. Different types of user billing ladder is k_1, k_2, k_3 three stages. Firstly, the initialization phase is carried out to complete the key negotiation and the allocation of the billing unit price $\omega_1 \xleftarrow{CRT} (1, 2, 3)$, $\omega_2 \xleftarrow{CRT} (0.3, 0.6, 1)$, $\omega_3 \xleftarrow{CRT} (0.5, 1, 1.5)$ corresponding to the ladder of different types of users. Secondly, the smart meter collected the data $m_1 \xleftarrow{CRT} (500, 600, 0)$, $m_2 \xleftarrow{CRT} (1000, 1500, 2000)$, $m_3 \xleftarrow{CRT} (200, 100, 0)$ corresponding to the ladder of the user in real-time, encrypted and processed by linear operation, and obtained $f(\omega, m)$ after aggregation. Finally, CC decrypts the total electricity charges of all users corresponding to the first, second, and third stairs. Based on the user's electricity consumption, the power company can obtain dynamic bills by comparing the total electric charge on each ladder with the initial set value, combining with the gradient electricity price, flexibly adjusting the charging strategy, saving energy, etc.

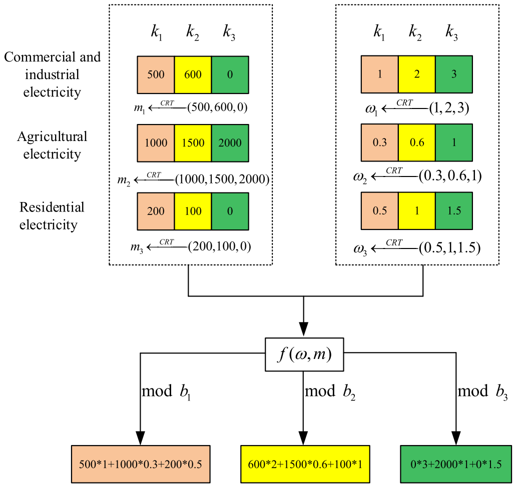

k_1 k_2 k_3
Commercial and industrial electricity
500 600 0
$m_1 \xleftarrow{CRT} (500, 600, 0)$
Agricultural electricity
1000 1500 2000
$m_2 \xleftarrow{CRT} (1000, 1500, 2000)$
Residential electricity
200 100 0
$m_3 \xleftarrow{CRT} (200, 100, 0)$
k_1 k_2 k_3
1 2 3
$\omega_1 \xleftarrow{CRT} (1, 2, 3)$
0.3 0.6 1
$\omega_2 \xleftarrow{CRT} (0.3, 0.6, 1)$
0.5 1 1.5
$\omega_3 \xleftarrow{CRT} (0.5, 1, 1.5)$
$f(\omega, m)$
mod b_1
mod b_2
mod b_3
500*1+1000*0.3+200*0.5
600*2+1500*0.6+100*1
0*3+2000*1+0*1.5

FIGURE 4. The step-by-step billing process.

VI. SCHEME ANALYSIS

A. SEMANTIC SECURITY OF ENCRYPTED DATA

In our scheme, each user encrypts the user's electricity consumption $c_i = g^{\omega_i \cdot m_i} H_2(T_s)^{N(sk_i+\beta_i)} \bmod N^2$ at the time T_s and submits it to FN for data aggregation, which we demonstrate to be semantically secure with the following theorem.

Lemma 1: For a given message m_0 or m_1, encrypted ciphers are indistinguishable.

Proof: Setup. The challenger obtains the parameters of the system. The attacker $\mathscr{A}$ obtains the public key of the system.

Ciphertext query. The attacker $\mathscr{A}$ inputs a plaintext message m_x, and the challenger returns the plaintext corresponding ciphertext c_x to the attacker $\mathscr{A}$. The attacker $\mathscr{A}$ can query the ciphertext corresponding to different plaintexts multiple times.

Challenge. When the query is finished. The attacker $\mathscr{A}$ sends two messages m_0 and m_1 to the challenger, where $|m_0| = |m_1|$. The challenger randomly selected $b \in \{0, 1\}$ and returns plain ciphertext pairs (c_b, m_b).

Guess. The attacker $\mathscr{A}$ outputs its guess $b' \in \{0, 1\}$. If $b' = b$. The attacker $\mathscr{A}$ wins the game.

The advantage of the attacker to win the game is defined as

$$Adv_{\mathscr{A}}^{\text{CPA}} = \left| p_r \left[b' = b \right] - \frac{1}{2} \right|$$

Defined. If for any polynomial-time attacker, there exists a negligible function $\varepsilon(\kappa)$ that makes $Adv_{\mathscr{A}}^{\text{CPA}}(\kappa) \leqslant \varepsilon(\kappa)$. The scheme is said to be semantically secure.

In our scheme, we randomly select a message in messages m_0 or m_1 by $c_b = g^{\omega_b \cdot m_b} H_2(T_s)^{N(sk_b+\beta_b)} \bmod N^2 (b \in \{0, 1\})$ to encrypt it. For the attacker $\mathscr{A}$, because the calculated

TABLE 3. Parameters of the type a curve.

Name	Value
q	8780710799663312522437781984754049815806883199414208211028653399266475630880222957078625179422662221423155858769582317459277713367317481324925129998224791
h	120160122648911460793888213667405342048029544012513118229196151310472072893597045311028448021839 06537786776
r	730750818665451621361119245571504901405976559617

$H_2(T_s)$ and $H_2(T_s) \overset{\$}{\leftarrow} \mathbb{Z}_{N^2}$ are indistinguishable. Refer to the indistinguishability of the paillier encryption scheme. $H_2(T_s)$ is equivalent to a randomly chosen $r(0 < r < N, r \in \mathbb{Z}^*_{N^2})$ in the paillier encryption scheme. Therefore, the indistinguishable ciphers $c_b(b \in \{0, 1\})$ and $c_b \overset{\$}{\leftarrow} \mathbb{Z}_{N^2}$ are also indistinguishable. The attacker $\mathscr{A}$ guess b' is a blind guess. Therefore, the attacker's advantage is $Adv^{\text{CPA}}_{\mathscr{A}} = \left|p_r\left[b' = b\right] - \frac{1}{2}\right| \leqslant \varepsilon(\kappa)$. In other words, the attacker $\mathscr{A}$ cannot distinguish messages m_0 and m_1, our scheme is semantically secure. End proof.

B. PRIVACY-PRESERVING

We analyzed our scheme of privacy protection through the following situation.

Case 1: Even if the attacker compromises the FN and obtains all messages sent by SM to the FN, the attacker still cannot obtain the sensitive information of a single user.

Analysis: The attacker destroys FN and obtains all the messages sent by SM to FN. Since the key of our scheme is generated through a secure key negotiation protocol, the attacker cannot infer the private key of any single user through this information. Because our scheme is semantically secure, no one can get a plaintext message through ciphertext without knowing the key. Therefore, even if the attacker destroys the FN, he cannot obtain any individual user's sensitive information.

Case 2: Even if the attacker has destroyed the CC, then an attacker can't get any single user's sensitive information.

Analysis: The attacker destroys the aggregated ciphertext obtained by the CC. The attacker can get the aggregated plaintext by decrypting it. Since our scheme is semantically secure, the attacker still cannot analyze any individual user's sensitive information through these plaintexts. Therefore, our scheme can protect the sensitive information of a single user.

Case 3: The attacker intentionally destroys some SM, FN cannot collect the data of these SM, the scheme uses the extended Shamir secret sharing scheme to achieve correct decryption, and the attacker cannot obtain any useful information in this process.

Analysis: We use the extended Shamir secret sharing scheme to achieve fault tolerance. Meanwhile, the user u_i divides the private key sk_i into two parts, one part is shared secretly and the other is saved by CC. In the process of secret reconstruction, after receiving the request, the user $u_v(v = 1, 2, 3, \cdots, \overline{n})$ calculates the equivalent ciphertext $c_{r,v} = H(T_s)^{y_{v,r}} \bmod Q$ of the secret share ($y_{v,r}$ represents the secret share of the user u_v sharing the faulty user u_r), and the original share is not exposed, so the secret share is reusable. At the same time, all the users in our scheme are semi-honest, so there is no collusion for k users of the refactoring. Therefore, no useful information is revealed during the refactoring process.

TABLE 4. Implementation parameters.

Parameter	Length
p, q	512 bits
ID_i, ID_f	32 bits
T, T_s	64 bits
H_1, H_2	1024 bits

TABLE 5. Time cost of related operations.

Notations	Runtime(ms)
T_e	28.41
T_m	0.13
T_a	0.04
T_σ	11.49
T_v	14.45
T_{h_1}	23.77
T_{h_2}	0.05

VII. PERFORMANCE EVALUATION

In this section, we evaluate our scenario in terms of computational overhead and communication overhead. Our experiment was based on the JPBC library, the computer used was configured with AMD R7-5800H CPU@3.2GHz and 16GB RAM, and the operating system was Windows 11. We use IntelliJ IDEA 2022.2.3(Community Edition), Open-JDK 64-Bit Server VM to run Java programs. The relevant parameters of bilinear mapping were shown in Table 3, and the sizes of selected parameters were shown in Table 4. We use T_e to represent the computing cost of exponential operation in the module N^2. T_m is used to represent the calculation cost of multiplication operation in the module N^2. T_a is used to represent the computing cost of exponential operation in the module N^2. T_σ is used to represent the computational cost of bilinear mapping signature. T_v represents the computational cost of bilinear mapping verification. T_{h_1} represents the computational cost of mapping $\{0, 1\}^*$ to G_1; T_{h_2} represents the computational overhead of mapping $\{0, 1\}^*$ to $Z_N{}^*$. The running times of these operations are shown in Table 5.

A. COMPUTATION OVERHEAD

In our scheme, the computing cost required by SM is $2T_e + T_m + T_\sigma + T_{h_1} + T_{h_2}$, FN is $nT_v + (n-1)T_m + T_\sigma + T_{h_1}$

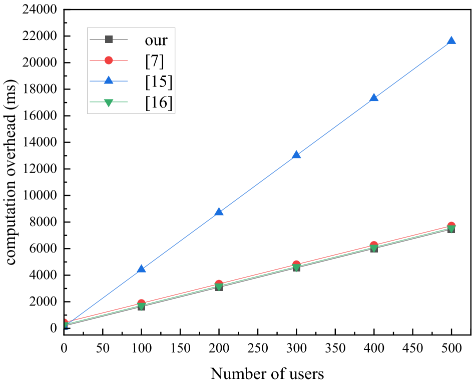


FIGURE 5. Comparison of computational overhead.

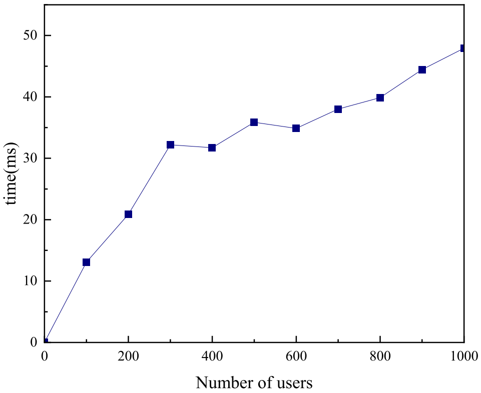


FIGURE 6. Computational overhead of shares generation for SMs with (13, 20).

and CC is $T_v + T_m + T_e + T_{h_2}$, so the total computing cost is $3T_e + (n+1)T_m + 2T_\sigma + (n+1)T_v + 2T_{h_1} + 2T_{h_2} = 14.58n + 170.43$ms.

In the scheme [7], the calculation cost in this scheme is related to the dimension. In order to facilitate calculation, the dimension is selected as 10 dimensions, and the calculation cost required by SM is $11T_e + 10T_m + T_\sigma + T_{h_1}$, $FN\ isnT_v + (n-1)T_m + T_\sigma + T_{h_1}$, and CC is $T_v + T_m + T_e$, so the total calculation cost is $12T_e + (n+10)T_m + 2T_\sigma + (n+1)T_v + 2T_{h_1} = 14.58n + 427.19$ms.

In the scheme [15], the calculation cost required by SM is $2T_m + T_a + T_e + T_\sigma + T_{h_1}$, FN is $nT_v + (n-1)T_e + (n-1)T_m + T_\sigma + T_{h_1}$, and CC is $T_v + T_m + T_e$, so the total calculation cost is $(n+1)T_e + (n+2)T_m + 2T_\sigma + (n+1)T_v + 2T_{h_1} + T_a = 42.99n + 113.68$ms.

In the scheme [16], considering that the energy efficiency level of common electrical appliances is usually level 4, the calculation cost required by SM is $4T_m + 5T_e + T_\sigma + T_{h_1}$, FN is $nT_v + (n-1)T_m + T_\sigma + T_{h_1}$ and CC is $T_v + T_m + T_e$, so the total calculation cost is $6T_e + (n+4)T_m + 2T_\sigma + (n+1)T_v + 2T_{h_1} = 14.58n + 255.95$ms.

As shown in Figure 5, we compare the computational overhead with other schemes, and our scheme has certain advantages in computational overhead. In addition, as shown in Figure 6, we test the computational overhead of share generation for SMs with (13, 20). As shown in Figure 7, when a part of the smart meter fails, we test the computational overhead of FN to recover the equivalent secret key using the extended Shamir secret sharing scheme.

B. COMMUNICATION OVERHEAD

The communication overhead is calculated based on the size of messages sent by the smart device to the fog node (SM-to-FN) and the fog node to the cloud center (FN-to-CC).

In our scheme, each user SM reports the message (c_i, σ_i, T, ID_i) to FN, so the communication overhead of SM-to-FN is $(|c_i| + |\sigma_i| + |T| + |ID_i|) = 3168n$ bits, FN sends data (c, σ_f, T, ID_f) to CC, so the communication overhead of FN-to-CC is $(|c| + |\sigma_f| + |T| + |ID_f|) = 3168$ bits.

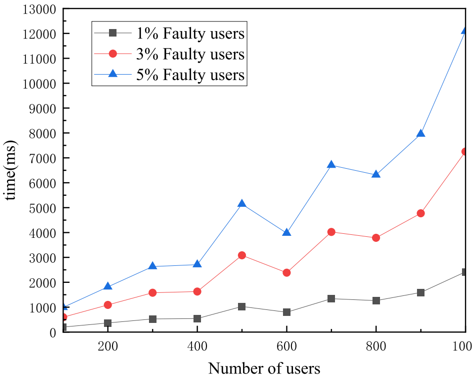


FIGURE 7. Computational overhead of reconstruction for FN with (13, 20).

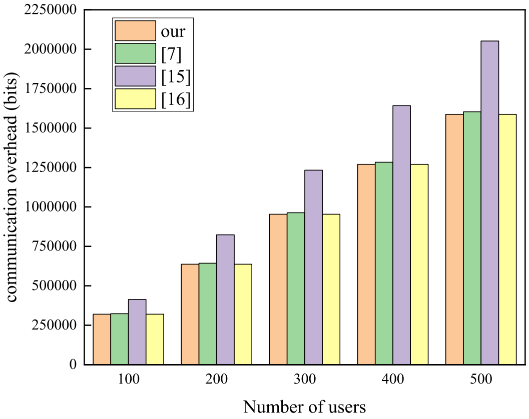


FIGURE 8. Comparison of communication overhead.

In the scheme [7], each user SM reports the message $(c_i, R_A, u_i, T_S, \sigma_i)$ to FN, so the communication overhead of SM-to-FN is $(|c_i| + |R_A| + |u_i| + |T_S| + |\sigma_i|) = 3200n$ bits, FN sends data $(C, R_A, G_W, T_S, \sigma_g)$ to CC, so the communication overhead of FN-to-CC is $(|C| + |R_A| + |G_W| + |T_S| + |\sigma_g|) = 3200$ bits.

In the scheme [15], each user SM reports the message (c_{ij}, k_{ij}, s_{ij}) to FN, so the communication overhead of

SM-to-FN is $(|c_{ij}| + |k_{ij}| + |s_{ij}|) = 4096n$ bits, FN sends data (c_i, k_i, s_i) to CC, so the communication overhead of FN-to-CC is $(|c_i| + |k_i| + |s_i|) = 4096$ bits.

In the scheme [16], each user SM reports the message $(c_{j,i}, s_i, t_i, tid_{sm_{i,q}})$ to FN, so the communication overhead of SM-to-FN is $(|c_{j,i}| + |s_i| + |t_i| + |tid_{sm_{i,q}}|) = 3168n$ bits, FN sends data $(c_j, s_j, t_j, ID_{FN_j})$ to CC, so the communication overhead of FN-to-CC is $(|c_j| + |s_j| + |t_j| + |ID_{FN_j}|) = 3168$ bits.

As shown in Figure 8, we compare the communication overhead with other schemes, and our scheme does not add an additional communication burden.

VIII. CONCLUSION

In this paper, we propose a fault-tolerant data aggregation scheme that supports the linear computation of multidimensional data without the participation of a trusted third party. The multi-dimensional data and weight of users are encoded by the Chinese residual theorem, which can improve the computational efficiency of encryption. In addition, we implement fault tolerance and user data integrity checking with an extended Shamir secret sharing scheme and bilinear mapping. The safety analysis results show that our scheme has a protective effect on the user's electricity consumption data. The experimental results show that our scheme is feasible and reasonable.

ACKNOWLEDGMENT

(Tanping Zhou and Zichao Song contributed equally to this work.)

REFERENCES

[1] S. M. Muyeen and S. Rahman, *Communication, Control and Security Challenges for the Smart Grid*. Rathipur, Odisha India: Institution of Engineering and Technology, 2017.

[2] N. S. Nafi, K. Ahmed, M. A. Gregory, and M. Datta, "A survey of smart grid architectures, applications, benefits and standardization," *J. Netw. Comput. Appl.*, vol. 76, pp. 23–36, Dec. 2016.

[3] S. K. Salman. (2017). *Introduction to the Smart Grid: Concepts, Technologies and Evolution*. [Online]. Available: https://www.iresearchbook.cn/f/ebook/detail?id=ec296388fec045faaec9ec0c2ec

[4] G. Dileep, "A survey on smart grid technologies and applications," *Renew. Energy*, vol. 146, pp. 2589–2625, Feb. 2020.

[5] M. Shateri, F. Messina, P. Piantanida, and F. Labeau, "Real-time privacy-preserving data release for smart meters," *IEEE Trans. Smart Grid*, vol. 11, no. 6, pp. 5174–5183, Nov. 2020.

[6] T. W. Chim, S. Yiu, V. O. K. Li, L. C. K. Hui, and J. Zhong, "PRGA: Privacy-preserving recording & gateway-assisted authentication of power usage information for smart grid," *IEEE Trans. Dependable Secure Comput.*, vol. 12, no. 1, pp. 85–97, Jan. 2015.

[7] R. Lu, X. Liang, X. Li, X. Lin, and X. Shen, "EPPA: An efficient and privacy-preserving aggregation scheme for secure smart grid communications," *IEEE Trans. Parallel Distrib. Syst.*, vol. 23, no. 9, pp. 1621–1631, Sep. 2012.

[8] S. Li, K. Xue, Q. Yang, and P. Hong, "PPMA: Privacy-preserving multisubset data aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 462–471, Feb. 2018.

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

[10] F. Knirsch, G. Eibl, and D. Engel, "Error-resilient masking approaches for privacy preserving data aggregation," *IEEE Trans. Smart Grid*, vol. 9, no. 4, pp. 3351–3361, Jul. 2018.

[11] X. Li, S. Liu, F. Wu, S. Kumari, and J. J. P. C. Rodrigues, "Privacy preserving data aggregation scheme for mobile edge computing assisted IoT applications," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4755–4763, Jun. 2019.

[12] X. Gong, Q. Hua, L. Qian, D. Yu, and H. Jin, "Communication-efficient and privacy-preserving data aggregation without trusted authority," in *Proc. IEEE INFOCOM Conf. Comput. Commun.*, Apr. 2018, pp. 1250–1258.

[13] Y. Liu, W. Guo, C. Fan, L. Chang, and C. Cheng, "A practical privacy-preserving data aggregation (3PDA) scheme for smart grid," *IEEE Trans. Ind. Informat.*, vol. 15, no. 3, pp. 1767–1774, Mar. 2019.

[14] Z. Song, W. Zhong, T. Zhou, D. Chen, Y. Ding, and X. Yang, "SEMDA: Secure and efficient multidimensional data aggregation in smart grid without a trusted third party," *Secur. Commun. Netw.*, vol. 2023, pp. 1–13, Feb. 2023.

[15] C. Peng, M. Luo, H. Wang, M. K. Khan, and D. He, "An efficient privacy-preserving aggregation scheme for multidimensional data in IoT," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 589–600, Jan. 2022.

[16] Z. Xia, Y. Zhang, K. Gu, X. Li, and W. Jia, "Secure multi-dimensional and multi-angle electricity data aggregation scheme for fog computing-based smart metering system," *IEEE Trans. Green Commun. Netw.*, vol. 6, no. 1, pp. 313–328, Mar. 2022.

[17] A. J. Menezes, P. C. Van Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 2018.

[18] P. Paillier, "Public-key cryptosystems based on composite degree residuosity classes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 1999, pp. 223–238.

[19] D. Boneh and M. Franklin, "Identity-based encryption from the Weil pairing," in *Proc. Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, 2001, pp. 213–229.

[20] Z. Liu, Z. Cao, X. Dong, X. Zhao, T. Liu, H. Bao, and J. Shen, "EPMDA-FED: Efficient and privacy-preserving multidimensional data aggregation scheme with fast error detection in smart grid," *IEEE Internet Things J.*, vol. 9, no. 9, pp. 6922–6933, May 2022.

[21] K. Xue, B. Zhu, Q. Yang, D. S. L. Wei, and M. Guizani, "An efficient and robust data aggregation scheme without a trusted authority for smart grid," *IEEE Internet Things J.*, vol. 7, no. 3, pp. 1949–1959, Mar. 2020.

[22] Z. Zeng, Y. Liu, and L. Chang, "A robust and optional privacy data aggregation scheme for fog-enhanced IoT network," *IEEE Syst. J.*, vol. 17, no. 1, pp. 1110–1120, Mar. 2023.

[23] O. R. Merad-Boudia and S. M. Senouci, "An efficient and secure multidimensional data aggregation for fog-computing-based smart grid," *IEEE Internet Things J.*, vol. 8, no. 8, pp. 6143–6153, Apr. 2021.

[24] L. Wu, M. Xu, S. Fu, Y. Luo, and Y. Wei, "FPDA: Fault-tolerant and privacy-enhanced data aggregation scheme in fog-assisted smart grid," *IEEE Internet Things J.*, vol. 9, no. 7, pp. 5254–5265, Apr. 2022.

[25] J. Liu, J. Weng, A. Yang, Y. Chen, and X. Lin, "Enabling efficient and privacy-preserving aggregation communication and function query for fog computing-based smart grid," *IEEE Trans. Smart Grid*, vol. 11, no. 1, pp. 247–257, Jan. 2020.

[26] L. Lyu, K. Nandakumar, B. Rubinstein, J. Jin, J. Bedo, and M. Palaniswami, "PPFA: Privacy preserving fog-enabled aggregation in smart grid," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3733–3744, Aug. 2018.

[27] X. Zuo, L. Li, H. Peng, S. Luo, and Y. Yang, "Privacy-preserving multidimensional data aggregation scheme without trusted authority in smart grid," *IEEE Syst. J.*, vol. 15, no. 1, pp. 395–406, Mar. 2021.

[28] A. Saleem, A. Khan, S. U. R. Malik, H. Pervaiz, H. Malik, M. Alam, and A. Jindal, "FESDA: Fog-enabled secure data aggregation in smart grid IoT network," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6132–6142, Jul. 2020.

[29] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[30] Y. Ming, X. Zhang, and X. Shen, "Efficient privacy-preserving multi-dimensional data aggregation scheme in smart grid," *IEEE Access*, vol. 7, pp. 32907–32921, 2019.

[31] Z. Zeng, X. Wang, Y. Liu, and L. Chang, "MSDA: Multi-subset data aggregation scheme without trusted third party," *Frontiers Comput. Sci.*, vol. 16, no. 1, pp. 1–7, Feb. 2022.

[32] F. Li, B. Luo, and P. Liu, "Secure information aggregation for smart grids using homomorphic encryption," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 327–332.

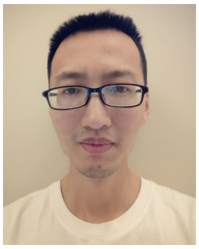
[33] O. R. M. Boudia, S. M. Senouci, and M. Feham, "Elliptic curve-based secure multidimensional aggregation for smart grid communications," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7750–7757, Dec. 2017.

ZICHAO SONG was born in Xi'an, China, in 1996. He is currently pursuing the master's degree with the Engineering University of People's Armed Police. His research interests include applied cryptography, cybersecurity, and privacy protection.

TANPING ZHOU was born in Yingtan, China, in 1989. He received the Ph.D. degree from the Engineering University of People's Armed Police. Currently, he is an Associate Professor at the Engineering University of People's Armed Police. His research interests include fully homomorphic encryption and encryption scheme based on lattice.

WEIDONG ZHONG was born in Ningxia, China, in 1970. He received the master's degree in software engineering from Chongqing University. He is currently a Professor with the Engineering University of People's Armed Police. His research interests include cryptography and network security.

DONG CHEN was born in Mianyang, China, in 1993. He is currently pursuing the master's degree with the Engineering University of People's Armed Police. His research interests include applied cryptography, cybersecurity, and privacy protection.



LONGFEI LIU was born in Zhoukou, China, in 1990. He is currently a Teaching Assistant at the Engineering University of People's Armed Police. His research interests include network security and stream cipher.

XIAOYUAN YANG was born in Xiangtan, China, in 1959. He is the Ph.D. Supervisor with the Engineering University of People's Armed Police. His research interests include information security and cryptology.