IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 3 May 2023, accepted 20 May 2023, date of publication 29 May 2023, date of current version 7 September 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3280637

RESEARCH ARTICLE

Privacy-Preserving Collaborative Sharing for Sharing Economy in Fog-Enhanced IoT

WENLE BAI AND AORAN HUANG
Institute of Information, North China University of Technology, Beijing 100144, China
Corresponding author: Aoran Huang (huang_aoran0519@163.com)

This work was supported in part by the Natural Science Foundation of Beijing under Grant 4212019 and Grant M22002.

ABSTRACT Fog-enhanced Internet of Things (IoT) has been widely deployed in the field of collaboration and sharing. However, participants expressed concern about the fairness of cost-sharing and privacy of data because of complicated collaborative sharing and untrusted fog nodes in network. In this paper, a novel privacy-preserving collaborative sharing protocol is proposed in fog-enhanced IoT. This protocol, based on the Paillier cryptosystems, can guarantee that only the coarse aggregate of users' requests are used to achieve fair cost-sharing without any communication between users. In addition, with the proposed protocol, the data stored in the device can be accurately transmitted to the user in accordance with each user's request without prying into the user's personal schedule. To demonstrate the security of our protocol, the thorough security analysis is performed. A significant number of experiments and comparison with existing schemes indicates that the suggested protocol is feasible.

INDEX TERMS Fog-enhanced Internet of Things, privacy-preserving, sharing economy, collaborative sharing, homomorphic encryption.

I. INTRODUCTION

Iot has been widely used in a variety of collaboration sharing fields [1], [2], such as smart city [3], knowledge management [4], smart contract [5]. During the collaborative process, participants are worried about the fairness of cost sharing throughout the partnership because it necessitates sharing a facility by several. A typical model can solve this problem of fairness called sharing economy, which enables users to share resources [6] and split costs in order to minimize user costs and optimize the utilization of IoT resources. In the sharing economy based on the IoT, there are a lot of users and connected devices, therefore network performance requirements are very strict. Fog computing emerges as a practical option for ensuring adequate computing power and low latency [7]. Fog servers are deployed to build the fog-enhanced IoT network [8] and used to aggregate and distribute the collected data in order to realize the sharing economy model in the IoT.

Despite fog computing can improve the performance of the sharing economy model in the IoT, the issue of privacy needs to be considered because of the untrusted fog servers [9]. Since user requests and device data must be processed by the fog servers, it is possible that the servers may purposefully leak this private information, causing the loss of user data and speculations about their activity [10]. To guarantee the security of the entire sharing economy process, a privacy-preserving protocol is required.

Recently, research on privacy protection in fog-enhanced IoT has primarily concentrated on geographical range query [11], [12], data aggregation schemes [13], [14], [15], and smart medicine [16]. Fairness has not been taken into account in these research due to limitations in processing power or communication delay. Studies on the sharing economy for fairness that use cloud computing are available [17], however because cloud servers are not as close to users as fog servers, it has much higher communication costs. Some research about fairness concentrates on resource distribution [7], [18], responsibility fairness [19], and preventing bias [20], but it is also restricted by network performance and ignores privacy concerns. Therefore, there is a dearth of research that takes both security and fairness into account.

The associate editor coordinating the review of this manuscript and approving it for publication was Chakchai So-In.

To address this problem, we propose an effective privacy-preserving protocol for sharing economy in fog-enhanced IoT. Only users, fog servers, and operators pose a threat to privacy in protocol. Users can resolve the issue of equitable cost-sharing based on this protocol and receive the information and services they are entitled to even if they only know their request. The Paillier cryptosystem's coarse-grained aggregation of privacy protection makes sure that fog servers and operators can only access the total number of requests for devices. Figure 1 displays the privacy-preserving sharing diagram of capacitated facility. The protocol focuses on sharing the data kept in the capacitated facility for the IoT device. For fog servers, only which devices are requested and the total number of requests can be known. For users, only the total number of requests for the devices they request can be known. To guarantee that the data can only be given to the users that request it, the threshold of whether the device replies is utilized to create a flag bit. Users can create their own distinctive random number vector to contaminate the data in response to threats from fog servers and operators in order to achieve the goal of preventing data leakage. In addition, our proposed protocol guarantees that computation expenses and communication overheads are both feasible and efficient. The purpose of this work is to solve the security problem when users collaborate and share public device in the fog-enhanced IoT. The novel protocol proposed can guarantee the safety of the sharing economy model in the context of fog-enhanced IoT. The three main contributions of this paper are as follows:

- It implements privacy-preserving fair cost-sharing. This protocol makes sure that fog servers and operators won't learn the precise request schedule of users for device sharing. Furthermore, the equitable sharing of sharing expenses can be accomplished with only coarse-grained request aggregation.
- Implemented capacitated facility privacy-preserving sharing. Without exposing the user request schedule, data stored in the capacity facilities can be securely and accurately transmitted to users for collaborative sharing.
- Provides rigorous security analysis of the protocol. The feasibility of the protocol is verified through experiments and comparisons in terms of computational cost and communication overhead.

The remainder of this paper is composed as follows. In Section II, related work is discussed. In Section III, it describes the model and design goals. In Section IV, some preparations for the Paillier cryptosystem are described. In Section V, the content and algorithm of the scheme are described in detail. In Section VI, the security performance of the scheme is analyzed. In Section VII, a performance evaluation was conducted. Paper is summarized in Section VIII.

II. RELATED WORK

A. PRIVACY-PRESERVING SYSTEM

The research of differential privacy (DP) in recent years mainly focuses on cyber physical systems [21], data statistics and machine learning. Zheng and Cai [22] Data analysis in industrial IoT is studied by using DP. Through perturbation mechanism, data in the IoT can be shared by consumers while protecting privacy of workers. Yang et al. [23] The combination of additional secret sharing and local differential privacy technologies can protect the geographic location of users when the user participates in crowdsourcing platforms. Wei et al. [24] studied federated learning with DP and expounded its efficiency and security. Liu et al. [25] proposed an architecture that can adapt to various secure multiparty machine learning tasks. Goyal and Saha [26] In resource constrained IoT systems, leverage the concurrent-transmission-based communication technology to efficiently realize a secure multi-party computation based strategy. The combination of homomorphic encryption (HE) and fog computing has drawn substantial attention to design of its mechanisms. Sendhil and Amuthan [27] discussed the advantages of HE when using fog computing for privacy protection aggregation, and discussed the application of HE in RSA, Paillier and other public key cryptosystems, as well as the requirements of fog computing for homomorphic encryption technology and the basic idea of HE [28]. Mahdikhani et al. [29] proposed an efficient privacy-preserving range query scheme using HE and reduced paths concept in fog-based IoT.

B. PRIVACY-PRESERVING TECHNIQUES IN FOG

There are studies on the total cost of users in community power system [6], but only the minimum total cost is studied, without discussing how to make participants share the cost fairly. Liu et al. [30] proposed a smart grid model based on fog computing, they proposed an efficient privacy protection scheme that supports aggregate communication and functional query. Lyu et al. [31] A privacy security aggregation scheme PPFA based on fog computing in smart grid is proposed, and the feasibility of the scheme is proved through data set testing in the real world. Li et al. [32] used MPC technology to propose a federated learning architecture for privacy-preserving under fog computing. Wei et al. [33] uses fog computing to build a privacy-preserving protocol for the vehicle group intelligent perception network, which reduces the computing cost and communication cost while ensuring security. Yekta and Lu [34], following the idea of XNOR logic gate, proposed a privacy protection query scheme for the fog-enhanced IoT, named XRQuery, which implements $O(\log n)$ between users and devices. Lu et al. [35] uses Paillier encryption, Chinese Remainder Theorem, and one way hash chain technologies to propose a lightweight privacy-preserving aggregation scheme in the fog-enhanced IoT.

Our proposed protocol, in contrast to the research mentioned above, focuses on achieving the privacy security of the sharing economy in fog-enhanced IoT. Each user can only access the data they specifically request, and device request fees can be fairly shared without revealing user privacy.

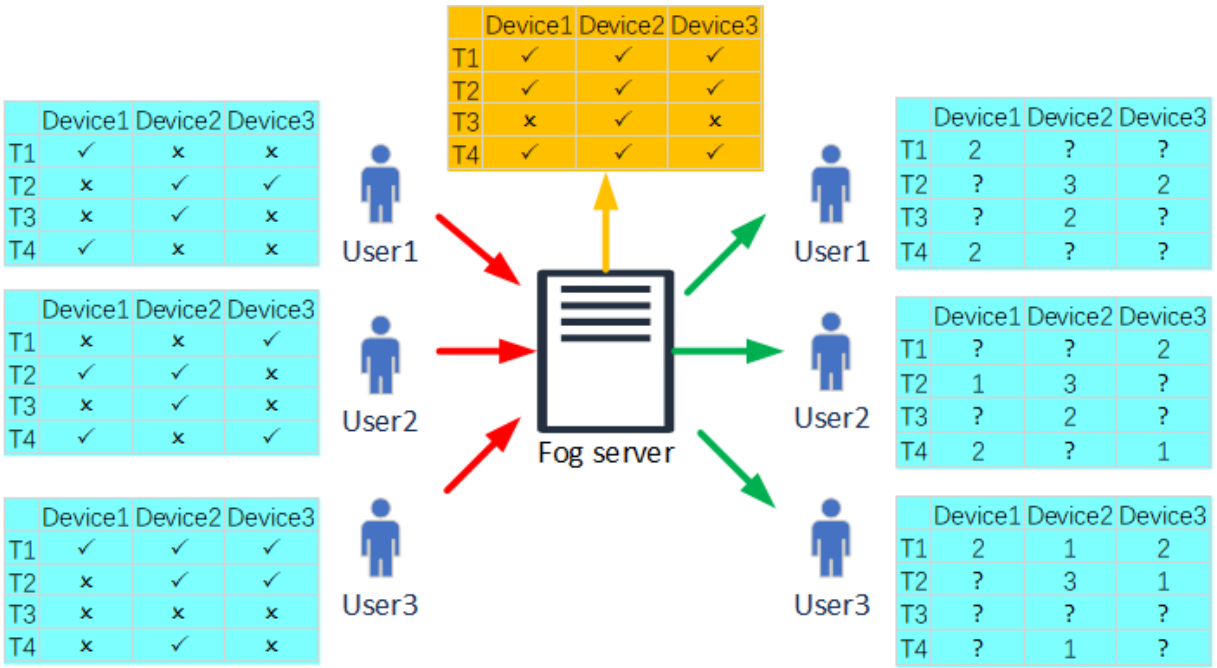

FIGURE 1. Illustrations of (a) privacy-preserving capacitated facility sharing.

TABLE 1. Key symbols and notations in the proposed scheme.

Notation	Description
FS1, FS2	Two fog servers FS1 and FS2 in the fog layer
U	The user set vector where $U = \{u_1, u_2, \cdots, u_k\}$
k	Number of users
I	The IoT devices Set Vector where $I = \{I_1, I_2, \cdots, I_N\}$
N	Number of devices
V_i	Vector of user i for IoT devices where $V_i = \{v_{i1}, v_{i2}, \cdots, v_{iN}\}$
P_i	Fees to be paid by user i
P_I	Request the cost vector of IoT n devices where $P_I = \{p_{I_1}, p_{I_2}, \cdots, p_{I_N}\}$
T_i	Coarse grained collection of devices requested by user i
$E()$	Paillier encryption function
$D()$	Paillier decryption function
C	Set vector of whether the device is requested where $C = \{c_1, c_2, \cdots, c_N\}$
D_n	Data contained in IoT device n
δ, ε	Two random numbers for contaminated data
$\sigma_{1\times k}, \lambda_{1\times k}$	Random number vector generated by user layer where $\sigma_{1\times k} = \{\sigma_1, \sigma_2, \cdots, \sigma_k\}, \lambda_{1\times k} = \{\lambda_1, \lambda_2, \cdots \lambda_k\}$
τ	Threshold to determine whether the device is requested
ϕ	Data range of IoT device D_i where $0 \leq D_i \leq \phi$
R_i	Data vector finally accepted by user i from FS1
R'_i	Data vector of plain text file finally obtained by user i
Q	Effective data vectors in IoT devices
Q_{con}	Q vector contaminated by random number
κ	Security parameter
$Z_{1\times N}$	Coarse aggregate vector requested by users for IoT devices where $Z_{1\times N} = \{z_1, z_2, \cdots, z_N\}$
(P_K, S_K)	The key pair of Paillier encryption

III. MODELS AND DESIGN GOAL

In this section, the security model and system architecture are given to further propose our design goals. The symbols used in this paper are shown in Table 1.

A. SECURITY MODEL

Before introducing the system architecture, the following four properties are considered in the scheme.

- Honest-but-curious Model: It assumes that all members in the system architecture are in line with the honest and curious model. They will not actively launch network attacks against others in the architecture, such as DOS attacks, but will only try to guess the privacy data of others through the results or data generated in the process of implementing the given privacy protection protocol.
- Without collusion: It assumes that each user and device does not communicate with each other. Users are not permitted to actively divulge their personal information to third parties or work together to guess their personal information.
- Privacy with operator/fog server: On the premise of providing services for users, operators/fog server can obtain as little private information as possible to ensure that they cannot understand the specific information of each user.
- Privacy with other users: For the purpose of using the requested devices, each user only receives the coarse aggregate rather than the fine aggregate in order to split the expense equally and keep their personal information private.

B. SYSTEM ARCHITECTURE

The architecture of the system is shown in Figure 1. The system is divided into four parts: user layer, fog layer, devices layer and operators.

- User Layer: Given the users vector $U = \{u_1, u_2, \cdots u_k\}$, it represents a total of k users. Exist $u_i \in U$, each u_i contains a request vector $V_i = \{v_{i1}, v_{i2}, \cdots, v_{iN}\}$ for IoT devices where $V_i \in u_i.N$ is the number of IoT devices and indicates the request of the i-th user to the

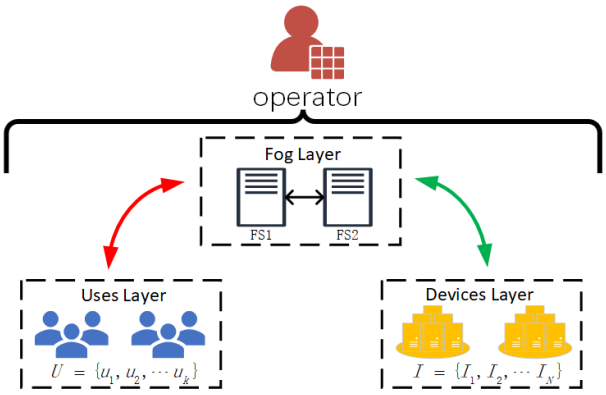

FIGURE 2. Schematic diagram of system architecture.

IoT device. $v_{in} \in \{0, 1\}$, 1 indicates that the user asks for the device; otherwise. The request vector V_i is only known by the user himself. Additionally, it is assumed that once the user sends a request to the device, all data in the device will be transmitted to the user.

- Device Layer: IoT device vector is given as $I = \{I_1, I_2, \cdots, I_N\}$,with N IoT devices in total. There is $D_n \in I_n$, which represents the data stored in the i-th IoT device and the upper limit of the size is ϕ, which is $0 \leq D_n \leq \phi$. There is a vector $Z_{1\times N}$ with the size of $1 \times N$, which represents the summary of the number of users who send requests to different IoT devices n. $P_I = \{p_{I_1}, p_{I_2}, \cdots, p_{I_N}\}$ exists, indicating the price required to request each device. τ represents the minimum number of requests responded by the IoT device. The pricing and τ of each device are publicly known to users and operators.
- Fog Server Layer:It consists of two fog servers FS1 and FS2, which communicate with users and device layer regularly. Because FS1 holds the private key S_K, it is the only member with decryption capability in the whole system, mainly responsible for the transmission with the user layer and sending information to the device layer. FS2 is mainly responsible for receiving information from user layer and device layer.
- Operator: It is deployed on the user layer, fog server layer and IoT device layer. The operator assigns the public key P_K of Paillier algorithm to the participants used, while the private key S_K is only sent to FS1. The operator is also responsible for accepting the fees paid by users at the user layer. After the collection is successful, the IoT device will provide services and data to users.

C. DESIGN GOAL

Our design goal is to solve the privacy security problem of the sharing economy in fog-enhanced IoT. This novel protocol also takes user privacy and security into account when prices and data are shared. There are specifically two goals:

- Fair Cost-Sharing with Privacy-Preserving: In the proposed protocol, users can accomplish fair cost-sharing through a privacy-preserving protocol that only allows them to know their own schedule, while servers or operators can only receive the overall number of facilities used.
- Capacitated Facility Sharing with Privacy-Preserving: The data on the device must be transferred through the server in order to complete the entire sharing process. Users can only comprehend the portions of the data matrix that pertain to their own requests when it is ultimately received, while servers and operators must deal with data that has been polluted throughout the whole transmission process.

IV. PRELIMINARIES

In this section, an overview of the Paillier Homomorphic Cryptosystem and Asymmetric Encryption System is presented. More specific introduction can be found in textbooks [36].

A. ASYMMETRIC ENCRYPTION SYSTEM

Different from symmetric encryption, in the asymmetric encryption system, encryption and decryption cannot be realized by only one key but requires two keys: public key P_K and private key S_K. The public key and private key appear in pairs, marked as (P_K, S_K). The plaintext to be transmitted is encrypted into ciphertext by public key P_K and then transmitted to the receiver. To decrypt the ciphertext, the S_K corresponding to the public key is used for decryption. In this paper, the encryption function is represented by $E(\bullet)$, and the decryption function is represented by $D(\bullet)$. If the information to be transmitted is x, formula (1) can been obtained.

$$\begin{aligned} E_{P_K}(x) &= y \\ D_{S_K}(y) &= x \end{aligned} \tag{1}$$

B. HOMOMORPHIC ENCRYPTION

Homomorphic encryption is a special encryption method that can operate on encrypted data, such as addition or multiplication. This feature ensures that data can be processed without data disclosure. After the receiver decrypts the result, the result is the result of data processing. Homomorphic encryption can be divided into three types. Partially Homomorphic Encryption (PHE): Only addition or multiplication is supported. Somewhat Homomorphic Encryption (SWHE): Both addition and multiplication operations are supported, but the number of operations is limited. Fully Homomorphic Encryption (FHE): Both addition and multiplication operations are supported, and the number of operations is unlimited.

C. PAILLIER CRYPTOSYSTEM

Paillier Cryptosystem is selected in our protocol, which is a typical asymmetric encryption system with homogeneity and is commonly used in privacy protection systems. It consists of the following three parts:

- Key generation algorithm $KeyGen(\varepsilon)$: Randomly select two large prime numbers p and q, satisfying $\gcd(pq, (p-1)(q-1)) = 1$. Note that the lengths of p and q

are equal to $|p| = |q| = \kappa$, called security parameters. Let $n = pq$ and $\lambda = lcm(p-1, q-1)$. Define function $L(x) = \frac{x-1}{n}$ and take integer $g \in \mathbb{Z}_{n^2}^*$, let $\mu = \left(L(g^\lambda \bmod n^2)\right)^{-1} \bmod n$. Finally, the public key $P_K = (n, g)$ and private key $S_K = (\lambda, \mu)$ can be obtained from the $KeyGen(\varepsilon)$ function.

- Encryption algorithm Encrypt():For plaintext m to be encrypted, $0 < m < n$ is required. Select random number r to satisfy $0 < r < n$ and $r \in \mathbb{Z}_{n^2}^*$. Then the ciphertext encrypted by public key P_K can be expressed $c = E(m) = g^m r^n \bmod n^2$.
- Decryption algorithm Decrypt():For the input ciphertext c, satisfy $c \in \mathbb{Z}_{n^2}^*$. Then get $m = D(c) = L(c^\lambda \bmod n^2) \cdot \mu \bmod n$ by using the private key decrypt.

It has the property of homomorphic addition and homomorphic multiplication, as shown in Formula (2) (3).

$$E(m_1) \cdot E(m_2) = E(m_1 + m_2) \tag{2}$$

$$E(m_1)^{m_2} = E(m_1 \cdot m_2) \tag{3}$$

The property of non-unique correspondence between ciphertext and plaintext is called self-blindness. It can be expressed as

$$D(E(m)r^n) = m \bmod n. \tag{4}$$

V. OUR PROPOSED PROTOCOL

This section provides a detailed overview of the privacy-preserving sharing economy protocol for fog-enhanced IoT. It is mainly divided into two parts: (1) fair sharing of costs; (2) data transmission of IoT devices.

A. FAIR SHARING OF COSTS

The protocol initially executes equitable cost sharing, and Figure 2 details the procedure.

1) SYSTEM INITIALIZATION

The safety parameters φ are given first, and the operator generates the public key and private key (P_K, S_K) through the $KeyGen(\varepsilon)$ function. After that, the operator sends the public key P_K to all members of the system. They can use the public key to encrypt their data. Only FS1 receives the private key S_K used for decryption.

In addition, random numbers δ, $\sigma_{1\times k}$ and $\lambda_{1\times k}$ for contaminated data are generated at the client. $\sigma_{1\times k}$ and $\lambda_{1\times k}$ are the set vector of k random numbers independently generated by users. Generate random number ε in FS2. Threshold τ is generated by the device and publicized to the user,and transmitted by the user to FS1. User vector $U = \{u_1, u_2, \cdots u_k\}$ and random number δ should first be encrypted before being transmitted to FS2. The data contained in FS2 contains encrypted user vectors, including $E(U) = \{E(u_1), E(u_2), \cdots, E(u_k)\}$ represents the encrypted user vector set, $E(u_i) = \{E(V_i)\}$ represents the encrypted i-th user feature set, $E(V_i) = \{E(v_{i1}), E(v_{i2}), \cdots, E(v_{iN})\}$ represents the encrypted i-th user request set for the IoT device, and the encrypted random number $E(\delta)$ is used for the subsequent data transmission process to contaminate the data. In Figure 2, this procedure is labeled as process ① and algorithm 1 briefly describes the implementation process.

Algorithm 1 Initialization

Input: $v_i \in U_i$
Output: $E(v_{in}), E(\delta)$
1: **for all** $v_i \in U_i$ **do**
2: $E(v_{in}) \leftarrow Paillier.En(v_{in})$
3: **end for**
4: $E(\delta) \leftarrow Paillier.En(\delta)$
5: **return** $E(v_{in}), E(\delta)$

2) AGGREGATION OF All REQUESTS

In order to determine which IoT devices users have requested overall, the request vector $V_i = \{v_{i1}, v_{i2}, \cdots, v_{iN}\}$ of the set of IoT devices that each user u_i in the user set $U = \{u_1, u_2, \cdots u_k\}$ requests must be added to obtain $\sum_{i=1}^{i=k} V_i$. In this process, everything is privacy-preserving. The specific steps are as follows:

The encrypted requests of all users are coarse-grained to obtain $Z_{1\times N}$ in FS2, as shown in (5).

$$\begin{aligned} |\, Z_{1\times N} &= \{z_1, z_2, \cdots, z_N\} \\ &= \left\{\prod_{i=1}^{k} E(v_{i1}), \prod_{i=1}^{k} E(v_{i2}), \cdots, \prod_{i=1}^{k} E(v_{iN})\right\} \\ &= \left\{E\left(\sum_{i=1}^{k} v_{i1}\right), E\left(\sum_{i=1}^{k} v_{i2}\right), \cdots, E\left(\sum_{i=1}^{k} v_{iN}\right)\right\} \end{aligned} \tag{5}$$

The aggregation of all users' encryption requests for IoT device n is $z_n = E(\sum_{i=1}^{k} v_{in})$. Transmitting $Z_{1\times N}$ to FS1 to decrypt can obtain

$$\begin{aligned} D(Z_{1\times N}) &= \{D(z_1), D(z_2), \cdots, D(z_N)\} \\ &= \left\{\sum_{i=1}^{k} v_{i1}, \sum_{i=1}^{k} v_{i2}, \cdots, \sum_{i=1}^{k} v_{iN}\right\}. \end{aligned} \tag{6}$$

It is necessary to judge whether the IoT device responds to determine whether the user needs to pay. Use $C = \{c_1, c_2, \cdots, c_N\}$ to represent the response set of devices, where

$$c_n = \begin{cases} 1 & \sum_{i=1}^{k} v_{in} - \tau > 0 \\ 0 & \sum_{i=1}^{k} v_{in} - \tau < 0 \end{cases} \tag{7}$$

Coarse aggregate after response judgment can be expressed as $\{c_1 \sum_{i=1}^{k} v_{i1}, c_2 \sum_{i=1}^{k} v_{i2}, \cdots, c_N \sum_{i=1}^{k} v_{iN}\}$, and send it to FS2.

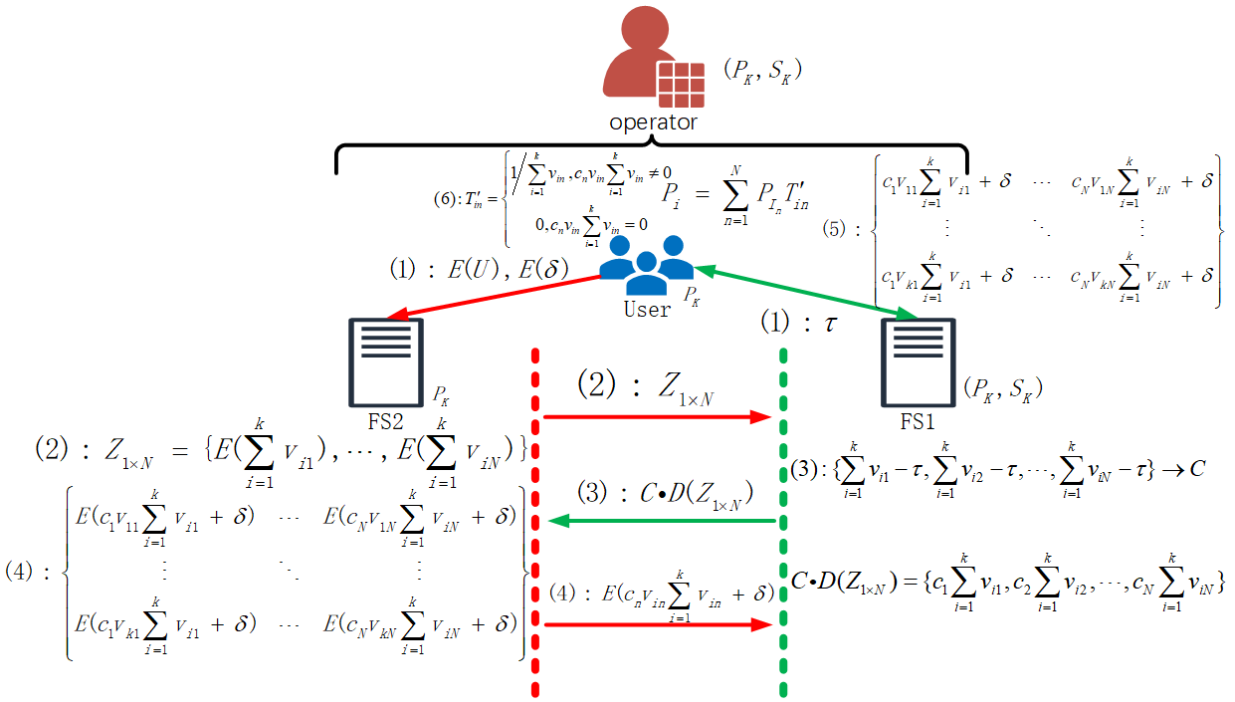

FIGURE 3. An illustration of fair sharing of costs, including system initialization, aggregation of all requests, coarse aggregate distribution and cost-sharing and payment.

In Figure 2, this procedure is labeled as process ② and ②. Algorithm 2 briefly describes the implementation process.

Algorithm 2 Requested Device Aggregation

//In FS2

for $i = 1$ to k **do**
  $Z_{1\times N}(i) = E(v_{i1})$
  for $n = 2$ to N **do**
    $Z_{1\times N}(i) \leftarrow Paillier.Add(E(v_{in}), Z_{1\times N}(i))$
  end for
end for
return $Z_{1\times N}$

//In FS1

for $i = 1$ to N **do**
  $D(Z_{1\times N}(i)) \leftarrow Paillier.De(Z_{1\times N}(i))$
end for
$C(i) \leftarrow D(Z(i)) - \tau \geq 0?1 : 0$
for $i = 1$ to N **do**
  $CD(i) \leftarrow C(i) \cdot Z_{1\times N}(i)$
end for
return CD

3) COARSE AGGREGATE DISTRIBUTION

In the last process, the coarse aggregates of all the requested IoT devices is obtained and stored in FS2. Next, need to assign them to users according to whether they request or not, to ensure that only when user i makes a request to device n, can user obtain the coarse granularity aggregation $c_n \sum_{i=1}^{k} v_{in}$ of the device's request, otherwise invalid data can be obtained. The specific steps in this stage are as follows: First of all, the encrypted user request a $E(v_{in})$ is operated with the power of $c_n \sum_{i=1}^{k} v_{in}$. Formula (8) can be obtained.

$$\begin{Bmatrix} (E(v_{11}))^{C_1 \sum_{i=1}^{k} v_{i1}} & \cdots & (E(v_{1N}))^{C_N \sum_{i=1}^{k} v_{iN}} \\ \vdots & \ddots & \vdots \\ (E(v_{k1}))^{C_1 \sum_{i=1}^{k} v_{i1}} & \cdots & (E(v_{kN}))^{C_N \sum_{i=1}^{k} v_{iN}} \end{Bmatrix} = \begin{Bmatrix} E(c_1 v_{11} \sum_{i=1}^{k} v_{i1}) & \cdots & E(c_N v_{1N} \sum_{i=1}^{k} v_{iN}) \\ \vdots & \ddots & \vdots \\ E(c_1 v_{k1} \sum_{i=1}^{k} v_{i1}) & \cdots & E(c_N v_{kN} \sum_{i=1}^{k} v_{iN}) \end{Bmatrix} \quad (8)$$

In order to prevent FS1 from directly obtaining the real $c_n v_{in} \sum_{i=1}^{k} v_{in}$ after decryption, it needs to use the encrypted random number $E(\delta)$ in FS2 to contaminate it. The process is as follows:

$$\begin{Bmatrix} E(c_1 v_{11} \sum_{i=1}^{k} v_{i1}) E(\delta) & \cdots & E(c_N v_{1N} \sum_{i=1}^{k} v_{iN}) E(\delta) \\ \vdots & \ddots & \vdots \\ E(c_1 v_{k1} \sum_{i=1}^{k} v_{i1}) E(\delta) & \cdots & E(c_N v_{kN} \sum_{i=1}^{k} v_{iN}) E(\delta) \end{Bmatrix}$$

Algorithm 3 Aggregate Assignment to Users

//In FS2
for $i = 1$ to k **do**
  for $n = 1$ to N **do**
    $Z(i) \leftarrow Paillier.Mul(E(v_{in}), D(Z_{1\times N}(n))$
    $Z(i) \leftarrow Paillier.Add(Z(i), E(\delta))$
  end for
end for
//In FS1
for $i = 1$ to k **do**
  $D(Z(i)) \leftarrow Paillier.De(Z(i))$
end for
//In User layer
for $i = 1$ to k **do**
  $T(i) \leftarrow D(Z(i)) - \delta$
end for
for $i = 1$ to k **do**
  for $n = 1$ to N **do**
    if $T(i, n) == 0$ **then**
      $T'(i, n) = 0$
    else
      $T'(i, n) = 1/T(i, n)$
    end if
  end for
end for

$$= \begin{Bmatrix} E(c_1 v_{11} \sum_{i=1}^{k} v_{i1} + \delta) & \cdots & E(c_N v_{1N} \sum_{i=1}^{k} v_{iN} + \delta) \\ \vdots & \ddots & \vdots \\ E(c_1 v_{k1} \sum_{i=1}^{k} v_{i1} + \delta) & \cdots & E(c_N v_{kN} \sum_{i=1}^{k} v_{iN} + \delta) \end{Bmatrix} \quad (9)$$

FS1 decrypts it to obtain $c_n v_{in} \sum_{i=1}^{k} v_{in} + \delta$, and then returns it to the user. Because the random number δ is generated at the client, it is known to the user. Therefore, the user i can obtain the coarse aggregate $T_i = (T_{i1}, T_{i2}, \cdots, T_{iN}) = \{c_1 v_{i1} \sum_{i=1}^{k} v_{i1}, c_2 v_{i2} \sum_{i=1}^{k} v_{i2}, \cdots, c_N v_{iN} \sum_{i=1}^{k} v_{iN}\}$ of the requested device by subtracting the random number from the received data. Therefore, user i can obtain the coarse aggregate of the requested devices by subtracting the random number δ from the received data. Since the request of user i for the device n is $v_{in} \in \{0, 1\}$ and $c_n \in \{0, 1\}$, the coarse aggregate $T_{in} \in \{0, \sum_{i=1}^{k} v_{in}\}$ obtained by it. If user i participates in the request for device n, the total number of users $\sum_{i=1}^{k} v_{in}$ participating in the request for device can be obtained. In Figure 2, this procedure is labeled as process ④ and ⑤. Algorithm 3 briefly describes the implementation process.

4) COST-SHARING AND PAYMENT

After the previous several stages, each user i gets a coarse-grained aggregate T_i of the number of devices he/she requests. For the convenience of calculation, it assumes that users' expenses for requesting device are evenly distributed according to the number of people. The price required for each device has been announced to the user in advance, which is $P_I = \{p_{I_1}, p_{I_2}, \cdots, p_{I_N}\}$. Take the reciprocal of each element in the vector T_i to get the vector T_i'. Therefore, the final fee payable by user i is

$$P_i = \sum_{n=1}^{N} P_{I_n} T'_{in}. \quad (10)$$

So far, fair cost sharing for privacy protection has been realized. In Figure 2, this procedure is labeled as process ⑥. After the user submits the fee, the system starts the data transmission part of the IoT device.

B. DATA TRANSMISSION OF IoT DEVICES

In this part, IoT device data must be securely transmitted to the user making the request. Since this part is a complete and continuous protocol with the part that fair sharing of costs, the system initialization is not required at the beginning, and the data stored in the above steps are still stored in FS1 and FS2. The data stored and used at this stage are the random numbers $\sigma_{1\times k}, \lambda_{1\times k}$ in the user layer. τ, C and $\{\sum_{i=1}^{k} v_{i1}, \sum_{i=1}^{k} v_{i2}, \cdots, \sum_{i=1}^{k} v_{iN}\}$ are in FS1. The random number ε and $E(V_i) = \{E(v_{i1}), E(v_{i2}), \ldots, E(v_{iN})\}$ are in FS2. Figure 3 shows the steps of data transmission process in detail.

1) DEVICES RESPONSE

It is necessary to determine which IoT devices are required. The random number $\sigma_{1\times k}$ in the user is transmitted to FS1, and the random number $\lambda_{1\times k}$ is encrypted and then transmitted to FS2. Set C is transmitted to the device layer and allocated to the corresponding device data to obtaine $C \cdot I = \{c_1 D_1, c_2 D_2, \cdots, c_N D_N\}$. Data D_n is stored in device n, so the requested data can be expressed as $c_n D_n$, where

$$c_n D_n = \begin{cases} D_n & c_n = 1 \\ 0 & c_n = 0 \end{cases} \quad (11)$$

Then the IoT device encrypts it with the public key P_K to obtain the effective data vector $Q = \{E(c_1 D_1), E(c_2 D_2), \cdots, E(c_N D_N)\}$ and transmits it to FS2. This step sets the data that does not need to be transmitted to 0 to avoid transmitting the data stored in the unresponsive device. In Figure 3, this procedure is labeled as process ① and ②. Algorithm 4 briefly describes this process.

Algorithm 4 Response of IoT Devices

for $i = 0$ to N **do**
  $C(n) \cdot D(n)$
  $Q(n) \leftarrow Paillier.En(C(n) \cdot D(n))$
end for
return Q

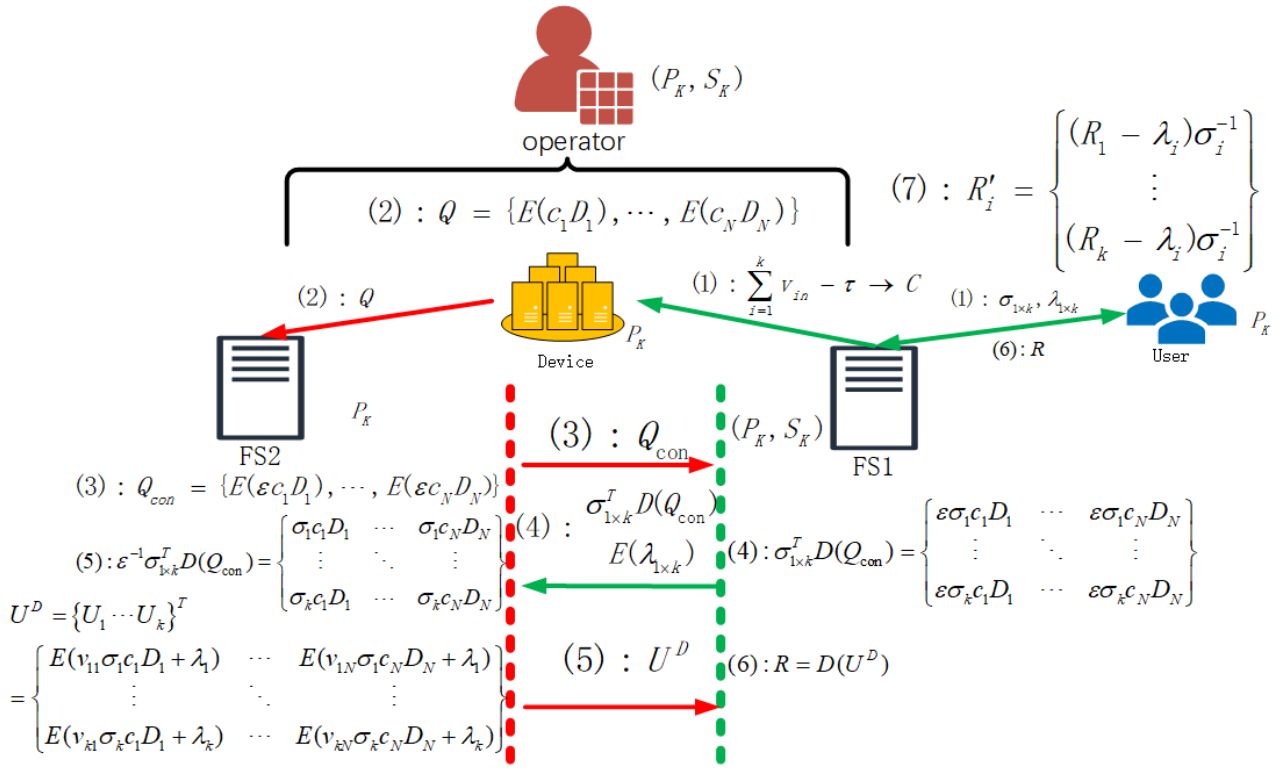

FIGURE 4. An illustration of data transmission in IoT devices, including judge whether the devices in the IoT respond, encrypted data processing in fog layer and data transmission to user layer.

2) ENCRYPTED DATA PROCESSING IN FOG LAYER

After receiving the data vector, FS2 must use random number ε to contaminate the data to stop FS1 from immediately decrypting the original data. So can obtain

$$\begin{aligned} Q_{con} &= \{E(c_1D_1)^{\varepsilon}, E(c_2D_2)^{\varepsilon}, \cdots, E(c_ND_N)^{\varepsilon}\} \\ &= \{E(\varepsilon c_1D_1), E(\varepsilon c_2D_2), \cdots, E(\varepsilon c_ND_N)\}. \end{aligned} \tag{12}$$

Q_{con} is transmitted to FS1 for decryption and obtain

$$D(Q_{con}) = \{\varepsilon c_1D_1, \varepsilon c_2D_2, \cdots, \varepsilon c_ND_N\}. \tag{13}$$

Data security in FS1 is guaranteed by the random number ε. Similarly, $D(Q_{con})$ must also be contaminated in FS1 before being spread to FS2. So the following can be obtained as

$$\sigma_{1\times k}^T D(Q_{con}) = \begin{Bmatrix} \varepsilon\sigma_1c_1D_1 & \cdots & \varepsilon\sigma_1c_ND_N \\ \vdots & \ddots & \vdots \\ \varepsilon\sigma_kc_1D_1 & \cdots & \varepsilon\sigma_kc_ND_N \end{Bmatrix} \tag{14}$$

$\sigma_{1\times k}^T D(Q_{con})$ is transmitted to FS2 and allocated to the specific user who sends the request. After FS2 receives $D(Q_{con})\sigma$, in order to save computing resources, first use ε^{-1} to eliminate a random number and obtain

$$\varepsilon^{-1}\sigma_{1\times k}^T D(Q_{con}) = \begin{Bmatrix} \sigma_1c_1D_1 & \cdots & \sigma_1c_ND_N \\ \vdots & \ddots & \vdots \\ \sigma_kc_1D_1 & \cdots & \sigma_kc_ND_N \end{Bmatrix}. \tag{15}$$

Then $E(V_i) = \{E(v_{i1}), E(v_{i2}), \ldots, E(v_{iN})\}$ stored in FS2 is used to distribute the data of the IoT device to users. For user i, according to the multiplicative homomorphism property, U_i^D can be obtained.

$$\begin{aligned} U_i^D &= \{E(v_{i1})^{\sigma_ic_1D_1}, E(v_{i2})^{\sigma_ic_2D_2}, \cdots, E(v_{iN})^{\sigma_ic_ND_N}\} \\ &= \{E(v_{i1}\sigma_ic_1D_1), E(v_{i2}\sigma_ic_2D_2), \cdots, E(v_{iN}\sigma_ic_ND_N)\} \end{aligned} \tag{16}$$

Because σ_i is a known random number by FS1, the data's privacy and security cannot be guaranteed. In order to protect the privacy of data, it is necessary to use random number λ_i to contaminate it.

$$\begin{aligned} U_i &= U_i^D E(\lambda_i) \\ &= \{E(v_{i1}\sigma_ic_1D_1), \cdots, E(v_{iN}\sigma_ic_ND_N)\}E(\lambda_i) \\ &= \{E(v_{i1}\sigma_ic_1D_1)E(\lambda_i), \cdots, E(v_{iN}\sigma_ic_ND_N)E(\lambda_i)\} \\ &= \{E(v_{i1}\sigma_ic_1D_1 + \lambda_i), \cdots, E(v_{iN}\sigma_ic_ND_N + \lambda_i)\} \end{aligned} \tag{17}$$

Therefore, the user's encrypted data matrix can be expressed as

$$\begin{aligned} U &= \{U_1 \cdots U_k\}^T \\ &= \begin{Bmatrix} E(v_{11}\sigma_1c_1D_1 + \lambda_1) & \cdots & E(v_{1N}\sigma_1c_ND_N + \lambda_1) \\ \vdots & \ddots & \vdots \\ E(v_{k1}\sigma_kc_1D_1 + \lambda_k) & \cdots & E(v_{kN}\sigma_kc_ND_N + \lambda_k) \end{Bmatrix} \end{aligned} \tag{18}$$

Finally, U is transferred to FS1. In Figure 3, this procedure is labeled as process ③, ④ and ⑤. Algorithm 5 briefly describes the implementation process.

Algorithm 5 Encrypted Data Processing

//In FS2
for $i = 0$ to N **do**
  $Qcon(i) \leftarrow Paillier.Mul(\varepsilon, Q(i))$
end for
//In FS1
for $i = 0$ to N **do**
  $DQcon(i) \leftarrow Paillier.De(Qcon(i))$
  $DQcon(i) \leftarrow Qcon(i) \cdot \sigma_i$
end for
//In FS2
for $i = 0$ to N **do**
  $DQcon(i) \leftarrow DQcon(i)/\varepsilon$
end for
for $i = 0$ to k **do**
  for $n = 0$ to N **do**
    $UD(i, n) \leftarrow Paillier.Mul(E(v_{in}), DQcon(n))$
    $UD(i, n) \leftarrow Paillier.Add(UD(i, n), E(\lambda_i))$
  end for
end for

Algorithm 6 Users Get Data

//In FS1
for $i = 0$ to k **do**
  for $n = 0$ to N **do**
    $R(i, n) \leftarrow Paillier.De(UD(i, n))$
    $R'(i, n) \leftarrow (R(i, n) - \lambda_i)/\sigma_i$
  end for
end for

3) DATA DISTRIBUTION AND RESTORE

U is decrypted in FS1 after receiving it.

$$R = D(U) = \{R_1, \cdots, R_i, \cdots, R_k\}^T = \begin{Bmatrix} v_{11}\sigma_1 c_1 D_1 + \lambda_1 & \cdots & v_{1N}\sigma_1 c_N D_N + \lambda_1 \\ \vdots & \ddots & \vdots \\ v_{k1}\sigma_k c_1 D_1 + \lambda_k & \cdots & v_{kN}\sigma_k c_N D_N + \lambda_k \end{Bmatrix} \tag{19}$$

For each element $v_{in}\sigma_i c_n D_n + \lambda_i$, exist $v_{in}\sigma_i c_n D_n + \lambda_i \in \{0, \sigma_i D_n + \lambda_i\}$ because of $v_{in} \in \{0, 1\}$ and $c_n \in \{0, 1\}$. If and only if user i makes a request to device n and this device responds, user i can receive the contaminated data $\sigma_i D_n + \lambda_i$. The random number σ_i, λ_i is generated at the user layer and is known to users. So after the user receives the data matrix and finds the corresponding row position, the set of IoT device data R'_i can be obtained.

$$R'_i = (R_i - \lambda_i){\sigma_i}^{-1} = \{v_{i1}c_1 D_1, v_{i2}c_2 D_2, \cdots, v_{iN}c_N D_N\} \tag{20}$$

where $v_{in}c_n D_n \in \{0, D_n\}$, D_n is taken when and only when user i sends a request to device n and device n responds. In Figure 3, this procedure is labeled as process ⑥ and ⑦. The implementation process is described in algorithm 6.

VI. SECURITY ANALYSIS

Based on the security model in section III, this part thoroughly examines the security of the protocol.

A. SECURITY AT THE COST CALCULATION STAGE

The user first sends the encrypted information $E(U), E(\delta)$ to FS2 and sends the threshold τ to FS1. FS2 only holds the public key P_K and cannot decrypt it, so this process is safe. Then obtain the aggregate set $Z_{1\times N} = E(\sum_{i=1}^{k} v_{i1}), E(\sum_{i=1}^{k} v_{i2}), \cdots, E(\sum_{i=1}^{k} v_{iN})\}$ of all users' requests for IoT devices by homomorphism property and transmit it to the fog server FS1. After decryption, FS1 obtains $D(Z_{1\times N}) = \{\sum_{i=1}^{k} v_{i1}, \sum_{i=1}^{k} v_{i2} \cdots, \sum_{i=1}^{k} v_{iN}\}$ and sends it to FS2. $\sum_{i=1}^{k} v_{in}$ indicates how many users have made requests to use IoT device n. Thus, FS1 and FS2 only know the total timetable of devices requests, but not the precise schedule of each user. It is also safe in this process. Upon the completion of the encryption calculation, the encryption vector $\{E(c_1 v_{i1} \sum_{i=1}^{k} v_{i1} + \delta), E(c_2 v_{i2} \sum_{i=1}^{k} v_{i2} + \delta), \cdots, E(c_N v_{iN} \sum_{i=1}^{k} v_{iN} + \delta)\}$ is obtained and sent to FS1 for decryption to produce $\{c_1 v_{i1} \sum_{i=1}^{k} v_{i1} + \delta, c_2 v_{i2} \sum_{i=1}^{k} v_{i2} + \delta, \cdots, c_N v_{iN} \sum_{i=1}^{k} v_{iN} + \delta\}$. For FS2, the encryption vector is unknown. For FS1, element $c_n v_{in} \sum_{i=1}^{k} v_{in} + \delta$ is contaminated by random number δ and is unable to extract actual information, resulting in privacy security. Then transmit it to the user layer. The user i receives $P_i = \sum_{n=1}^{N} P_{I_n} T'_{in}$, indicating the fee he/she must pay. For devices not requested by the user, there is no fee.

B. SECURITY DURING DATA TRANSMISSION

Assess the device's response. Just the devices' response can be obtained in step FS1, not the user's precise timetable. Devices process their data and produces $Q = \{E(c_1 D_1), E(c_2 D_2), \cdots, E(c_N D_N)\}$ depending on how it responds. After receiving the encrypted data, FS2 pollutes it with random numbers ε before sending it to FS1. Due to the presence of unknown random numbers, FS1 is unable to determine the genuine information for the decrypted $D(Q_{con})$, assuring privacy security. Since that FS2 already knows the random number ε, using $\sigma^T_{1\times k}$ to contaminate it before

TABLE 2. The experiment parameter settings.

Parameter	Meaning description
κ	$\kappa = \{256, 512, 1024, 2048, 3072\}$: security parameter
k	$k = \{5, 10, 15, 20, 25, 30\}$:Number of users
N	$N = \{10, 20, 30, 40, 50\}$:Number of IoT devices
η	$0 \leq \eta \leq 1$, proportion of IoT devices requested
L	Length of ciphertext encrypted by Pailier

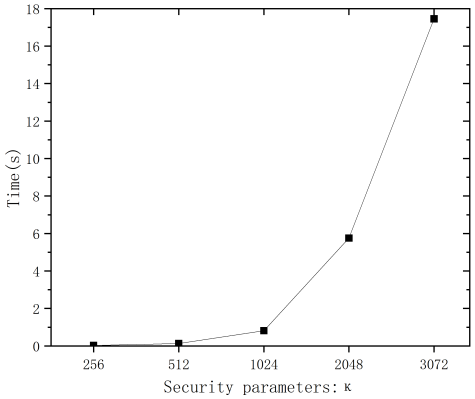

FIGURE 5. The running time required for the initialization process in the scheme under different security parameter.

transmission to FS2 is required. Similarly, FS2 is unable to collect accurate information of $\varepsilon^{-1}\sigma_{1\times k}^{T} D(Q_{con})$. The matrix U^D is obtained after data distribution, and each of its component parts contains a random number λ_i. Following FS1's decryption, an unknown λ_i for FS1 is used to assure the security of the matrix. Each user can only recover the data they are supposed to receive after transmitting U^D to the user layer because they only have the independently generated random numbers σ_i and λ_i in their possession. In addition, users can only determine the location of the data that needs to be decrypted by themselves, without knowing the location of others. It ensures the security of its own data.

VII. PERFORMANCE EVALUATION

This section analyzes the performance of the proposed protocol in terms of communication and computing costs. Each experiment was carried out 20 times, with the findings averaged. Table 2 lists the precise parameter settings used in the experiment.

A. COMPUTATIONAL COSTS

Figure 4 illustrates the changing trend in system initialization time when the security parameter is $\kappa = 256, 512, 1024, 2048, 3072$ for a scenario with 5 users and 100 devices. It is clear that as the safety parameter increases, so does the time required for system initialization. The safety parameter is set to $\kappa = 2048$ in following experiments.

For the five different algorithms requested device aggregation, aggregate assignment to users, response of IoT devices, encrypted data processing and users get data, the computational cost is considered from three aspects. The effect of the request proportion on the calculation cost is depicted in Figure 5. As can be seen, the calculation cost is unaffected by the request proportion. This makes sense given that the gadget must take part in the calculation even if it is not requested.

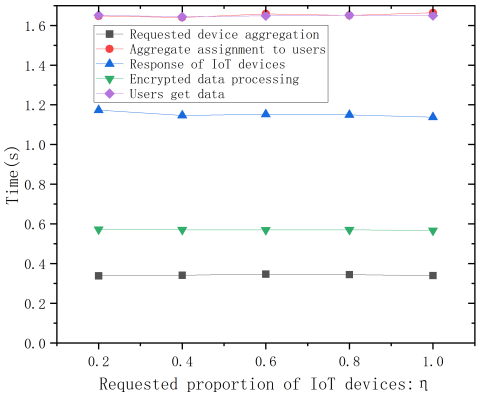

FIGURE 6. The running time of the five processes under different requested promotion of IoT devices, including requested device aggregation, aggregate assignment to users, response of IoT devices, encrypted data processing and users get data.

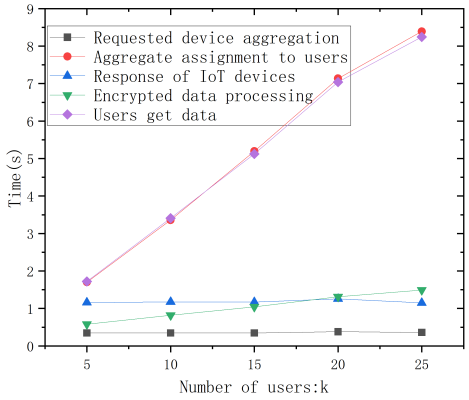

FIGURE 7. The running time of the five processes under different number of users, including requested device aggregation, aggregate assignment to users, response of IoT devices, encrypted data processing and users get data.

Figure 6 displays the five algorithms' execution times for user counts $k = 5, 10, 15, 20, 25$. The number of users has a positive correlation with the running times of algorithms aggregate assignment to users, encrypted data processing and users get data, whereas requested device aggregation and response of IoT devices have no such correlation, because devices are the focus of the latter two activities. Figure 7 demonstrates a positive association between the running times of the five algorithms and the number of devices at the device counts $N = 100, 200, 300, 400, 500$.

The cost estimates for the two stages fair sharing of costs and data transmission of IoT devices in the protocol are contrasted with the computational cost for the PPSA [14] on the assumption of 5 users and 100 devices. Figure 8 demonstrates that the computation cost for stage fair sharing and data transmission are both lower than the PPSA's. It can be confirmed that the computational cost of our protocol is feasible.

B. COMMUNICATION OVERHEADS

First, analyze the communication overhead generated in the phase of determining the amount that users need to pay, and

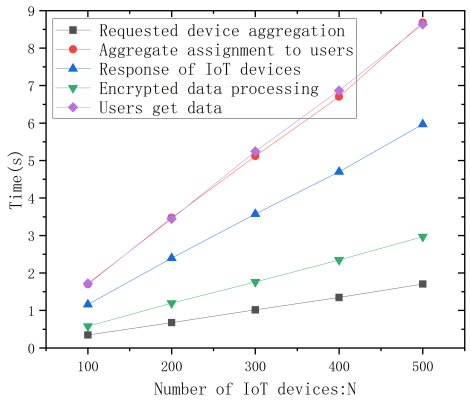

FIGURE 8. The running time of the five processes under different number of IoT devices, including requested device aggregation, aggregate assignment to users, response of IoT devices, encrypted data processing and users get data.

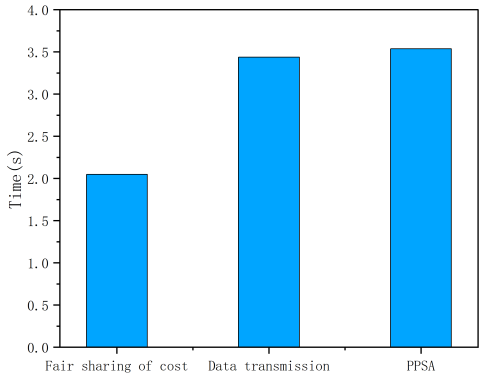

FIGURE 9. A comparison of the computational costs for stage fair sharing of costs and data transmission of IoT devices with scheme PPSA.

TABLE 3. The communication overhead of determining the amount to be paid by the user.

Step	Transmission route	Overheads
(1)	U → FS1, FS2	$k \cdot N \cdot L + L + 1$
(2)	FS2 → FS1	$N \cdot L$
(3)	FS1 → FS2	N
(4)	FS2 → FS1	$k \cdot N \cdot L$
(5)	FS2 → U	$k \cdot N$

summarize them in Table 3. The user sends the encrypted user vector $E(U)$ and the encrypted random number $E(\sigma)$ to FS2,threshold τ to FS1. Each user vector U also includes N device request vectors, so the communication overhead generated by this process is $k \cdot N \cdot L + L + 1$, where L represents the length of ciphertext after Paillier encryption, k represents the number of users, and N represents the number of devices. After the data is processed in FS2, $Z_{1\times N}$ is sent to FS1. The communication overhead incurred in this process is $N \cdot L$. Decrypt in FS1 and return it to FS2. The overhead of this process is N. Next, $E(c_n v_{in} \sum_{i=1}^{k} v_{in} + \delta)$ is obtained from FS2 and transmitted to FS1, resulting in a communication overhead of $k \cdot N \cdot L$. Finally, FS1 decrypts and transmits the decrypted data to the user. The communication overhead incurred by this process is $k \cdot N$.

TABLE 4. The communication overheads of data transmission in IoT devices.

Step	Transmission route	Overheads
(1)	FS1 → I, U → FS1	$N + 2 \cdot k$
(2)	I → FS2	$N \cdot L$
(3)	FS2 → FS1	$N \cdot L$
(4)	FS1 → FS2	$2 \cdot k \cdot N$
(5)	FS2 → FS1	$k \cdot N \cdot L$
(6)	FS1 → U	$k \cdot N$

Secondly, analyze the communication overhead generated in the data transmission phase of the IoT device and summarize them in Table 4. FS1 sends C to the IoT device end and user layer sends random number $\sigma_{1\times k}, \lambda_{1\times k}$ to FS1. Therefore, the communication overhead is $N + 2 \cdot k$. Then the IoT device end sends the encrypted IoT device data Q to FS2, resulting in communication overhead $N \cdot L$. After FS2 processes the data, Q_{con} sends it to FS1, resulting in a communication overhead of $N \cdot L$. Then FS1 decrypts it and transmits it back to FS2 after processing. The communication cost incurred in this process is $2 \cdot k \cdot N$. After processing in FS2, $E(v_{in}\sigma_i c_n D_n + \lambda_i)$ is obtained and transmitted to FS1. The transmission communication overhead is $k \cdot N \cdot L$. Finally, after decryption by FS1, the obtained data is transmitted to the user and the communication overhead $k \cdot N$ is generated. Based on the above discussion, it can be concluded that the communication overheads of our protocol is acceptable.

VIII. CONCLUSION

This paper presents the work of privacy-servicing protocol for sharing economy in fog-enhance IoT, which considers both fair cost sharing and data privacy protection. It realizes the security of user information and device data in the process of user request, IoT device response and data transmission. Users can pay the required fees fairly according to the number of people requesting IoT devices and share the data stored in the requested IoT devices. A thorough investigation was done to make sure the protocol safeguarded privacy. According to the findings of the protocol performance testing, the protocol is feasible in terms of computational and communication overhead. Next, this work will be applied to real-world scenario, such as smart grids, smart cities, etc.

REFERENCES

[1] T. Cai, W. Chen, K. E. Psannis, S. K. Goudos, Y. Yu, Z. Zheng, and S. Wan, "Scalable on-chain and off-chain blockchain for sharing economy in large-scale wireless networks," *IEEE Wireless Commun.*, vol. 29, no. 3, pp. 32–38, Jun. 2022.

[2] Y. Liu, K. Xue, P. He, D. S. L. Wei, and M. Guizani, "An efficient, accountable, and privacy-preserving access control scheme for Internet of Things in a sharing economy environment," *IEEE Internet Things J.*, vol. 7, no. 7, pp. 6634–6646, Jul. 2020.

[3] Md. A. Rahman, Md. M. Rashid, M. S. Hossain, E. Hassanain, M. F. Alhamid, and M. Guizani, "Blockchain and IoT-based cognitive edge framework for sharing economy services in a smart city," *IEEE Access*, vol. 7, pp. 18611–18621, 2019.

[4] M. F. Manesh, M. M. Pellegrini, G. Marzi, and M. Dabic, "Knowledge management in the fourth industrial revolution: Mapping the literature and scoping future avenues," *IEEE Trans. Eng. Manag.*, vol. 68, no. 1, pp. 289–300, Feb. 2021.

[5] M. N. Islam and S. Kundu, "Poster abstract: Preserving IoT privacy in sharing economy via smart contract," in *Proc. IEEE/ACM 3rd Int. Conf. Internet Things Design Implement. (IoTDI)*, Apr. 2018, pp. 296–297.
[6] M. Nabil, A. Sherif, M. Mahmoud, A. Alsharif, and M. Abdallah, "Efficient and privacy-preserving ridesharing organization for transferable and non-transferable services," *IEEE Trans. Depend. Secure Comput.*, vol. 18, no. 3, pp. 1291–1306, May 2021.
[7] R. Bukhsh, N. Javaid, R. A. Abbasi, A. Fatima, M. Akbar, M. K. Afzal, and F. Ishmanov, "An efficient fog as-a-power-economy-sharing service," *IEEE Access*, vol. 7, pp. 185012–185027, 2019.
[8] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 1, pp. 416–464, 1st Quart., 2018.
[9] N. Neshenko, E. Bou-Harb, J. Crichigno, G. Kaddoum, and N. Ghani, "Demystifying IoT security: An exhaustive survey on IoT vulnerabilities and a first empirical look on internet-scale IoT exploitations," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2702–2733, 3rd Quart., 2019.
[10] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang, and W. Zhao, "A survey on Internet of Things: Architecture, enabling technologies, security and privacy, and applications," *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1125–1142, Oct. 2017.
[11] Y. Guo, H. Xie, C. Wang, and X. Jia, "Enabling privacy-preserving geographic range query in fog-enhanced IoT services," *IEEE Trans. Depend. Secure Comput.*, vol. 19, no. 5, pp. 3401–3416, Sep. 2022.
[12] H. Xie, Y. Guo, and X. Jia, "Privacy-preserving location-based data queries in fog-enhanced sensor networks," *IEEE Internet Things J.*, vol. 9, no. 14, pp. 12285–12299, Jul. 2022.
[13] R. Lu, "A new communication-efficient privacy-preserving range query scheme in fog-enhanced IoT," *IEEE Internet Things J.*, vol. 6, no. 2, pp. 2497–2505, Apr. 2019.
[14] H. Mahdikhani, S. Mahdavifar, R. Lu, H. Zhu, and A. A. Ghorbani, "Achieving privacy-preserving subset aggregation in fog-enhanced IoT," *IEEE Access*, vol. 7, pp. 184438–184447, 2019.
[15] H. Mahdikhani, R. Lu, Y. Zheng, and A. Ghorbani, "Achieving efficient and privacy-preserving range query in fog-enhanced IoT with Bloom filter," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 1–6.
[16] R. Saha, G. Kumar, M. K. Rai, R. Thomas, and S.-J. Lim, "Privacy ensured *e*-healthcare for fog-enhanced IoT based applications," *IEEE Access*, vol. 7, pp. 44536–44543, 2019.
[17] L. Lyu, S. C. Chau, N. Wang, and Y. Zheng, "Cloud-based privacy-preserving collaborative consumption for sharing economy," *IEEE Trans. Cloud Comput.*, vol. 10, no. 3, pp. 1647–1660, Jul. 2022.
[18] L. C. C. De Biase, P. C. Calcina-Ccori, G. Fedrecheski, G. M. Duarte, P. S. S. Rangel, and M. K. Zuffo, "Swarm economy: A model for transactions in a distributed and organic IoT platform," *IEEE Internet Things J.*, vol. 6, no. 3, pp. 4561–4572, Jun. 2019.
[19] S. K. Lo, Y. Liu, Q. Lu, C. Wang, X. Xu, H.-Y. Paik, and L. Zhu, "Toward trustworthy AI: Blockchain-based architecture design for accountability and fairness of federated learning systems," *IEEE Internet Things J.*, vol. 10, no. 4, pp. 3276–3284, Feb. 2023.
[20] Á. González, F. Ortega, D. Pérez-López, and S. Alonso, "Bias and unfairness of collaborative filtering based recommender systems in MovieLens dataset," *IEEE Access*, vol. 10, pp. 68429–68439, 2022.
[21] M. U. Hassan, M. H. Rehmani, and J. Chen, "Differential privacy techniques for cyber physical systems: A survey," *IEEE Commun. Surveys Tuts.*, vol. 22, no. 1, pp. 746–789, 1st Quart., 2020.
[22] X. Zheng and Z. Cai, "Privacy-preserved data sharing towards multiple parties in industrial IoTs," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 5, pp. 968–979, May 2020.
[23] M. Yang, I. Tjuawinata, K. Y. Lam, J. Zhao, and L. Sun, "Secure hot path crowdsourcing with local differential privacy under fog computing architecture," *IEEE Trans. Services Comput.*, vol. 15, no. 4, pp. 2188–2201, Jul. 2022.
[24] K. Wei, J. Li, M. Ding, C. Ma, H. H. Yang, F. Farokhi, S. Jin, T. Q. S. Quek, and H. V. Poor, "Federated learning with differential privacy: Algorithms and performance analysis," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 3454–3469, 2020.
[25] Y. Liu, Y. Kang, C. Xing, T. Chen, and Q. Yang, "A secure federated transfer learning framework," *IEEE Intell. Syst.*, vol. 35, no. 4, pp. 70–82, Jul. 2020.
[26] H. Goyal and S. Saha, "Multi-party computation in IoT for privacy-preservation," in *Proc. IEEE 42nd Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jul. 2022, pp. 1280–1281.
[27] R. Sendhil and A. Amuthan, "Privacy preserving data aggregation in fog computing using homomorphic encryption: An analysis," in *Proc. Int. Conf. Comput. Commun. Informat. (ICCCI)*, Jan. 2020, pp. 1–5.
[28] R. Sendhil and A. Amuthan, "A descriptive study on homomorphic encryption schemes for enhancing security in fog computing," in *Proc. Int. Conf. Smart Electron. Commun. (ICOSEC)*, Sep. 2020, pp. 738–743.
[29] H. Mahdikhani, R. Lu, J. Shao, and A. Ghorbani, "Using reduced paths to achieve efficient privacy-preserving range query in fog-based IoT," *IEEE Internet Things J.*, vol. 8, no. 6, pp. 4762–4774, Mar. 2021.
[30] J.-N. 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.
[31] 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.
[32] Y. Li, H. Li, G. Xu, T. Xiang, and R. Lu, "Practical privacy-preserving federated learning in vehicular fog computing," *IEEE Trans. Veh. Technol.*, vol. 71, no. 5, pp. 4692–4705, May 2022.
[33] J. Wei, X. Wang, N. Li, G. Yang, and Y. Mu, "A privacy-preserving fog computing framework for vehicular crowdsensing networks," *IEEE Access*, vol. 6, pp. 43776–43784, 2018.
[34] N. I. Yekta and R. Lu, "XRQuery: Achieving communication-efficient privacy-preserving query for fog-enhanced IoT," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.
[35] R. Lu, K. Heung, A. H. Lashkari, and A. A. Ghorbani, "A lightweight privacy-preserving data aggregation scheme for fog computing-enhanced IoT," *IEEE Access*, vol. 5, pp. 3302–3312, 2017.
[36] X. Yi, R. Paulet, and E. Bertino, *Homomorphic Encryption and Applications*. Berlin, Germany: Springer, 2014.

WENLE BAI was born in Shanxi, China, in 1967. He received the Ph.D. degree in communication engineering from the Beijing University of Posts and Telecommunication, China, in 2006. He is currently a Professor with the North China University of Technology. He has published about 20 articles in the related areas. His research interests include apply cryptography and information security.

AORAN HUANG was born in Liaoning, China. He received the B.E. degree in electronic information engineering from the College of Information Science and Technology, North China University of Technology, in 2021, where he is currently pursuing the M.E. degree in information and communication engineering. His research interests include fog-enhanced IoT, fairness, and data privacy.