computers

MDPI

Article

An Efficient Attribute-Based Participant Selecting Scheme with Blockchain for Federated Learning in Smart Cities

Xiaojun Yin, Haochen Qiu, Xijun Wu and Xinming Zhang *

School of Computer Science and Technology, University of Science and Technology of China, Hefei 230027, China; 13911566888@139.com (X.Y.); qhc1997@mail.ustc.edu.cn (H.Q.); wuxijun@mail.ustc.edu.cn (X.W.)
* Correspondence: xinming@ustc.edu.cn

Abstract: In smart cities, large amounts of multi-source data are generated all the time. A model established via machine learning can mine information from these data and enable many valuable applications. With concerns about data privacy, it is becoming increasingly difficult for the publishers of these applications to obtain users' data, which hinders the previous paradigm of centralized training through collecting data on a large scale. Federated learning is expected to prevent the leakage of private data by allowing users to train models locally. The existing works generally ignore architectures designed in real scenarios. Thus, there still exist some challenges that have not yet been explored in federated learning applied in smart cities, such as avoiding sharing models with improper parties under privacy requirements and designing satisfactory incentive mechanisms. Therefore, we propose an efficient attribute-based participant selecting scheme to ensure that only someone who meets the requirements of the task publisher can participate in training under the premise of high privacy requirements, so as to improve the efficiency and avoid attacks. We further extend our scheme to encourage clients to take part in federated learning and provide an audit mechanism using a consortium blockchain. Finally, we present an in-depth discussion of the proposed scheme by comparing it to different methods. The results show that our scheme can improve the efficiency of federated learning by enabling reliable participant selection and promote the extensive use of federated learning in smart cities.

Keywords: smart cities; blockchain; CP-ABE; federated learning

Citation: Yin, X.; Qiu, H.; Wu, X.; Zhang, X. An Efficient Attribute-Based Participant Selecting Scheme with Blockchain for Federated Learning in Smart Cities. *Computers* **2024**, *13*, 118. https://doi.org/10.3390/computers13050118

Academic Editors: Caterina Tricase, Otar Zumburidze, Nino Adamashvili, Radu State and Roberto Tonelli

Received: 8 April 2024
Revised: 30 April 2024
Accepted: 30 April 2024
Published: 9 May 2024

1. Introduction

The concept of smart cities has become central to contemporary discussions on urban development, where the integration of Information and Communication Technology (ICT) is pivotal in transforming the city's infrastructure and services [1,2]. Smart cities utilize advanced data analytics and IoT technologies to optimize resources, improve service delivery, and enhance the quality of urban life. These urban areas are defined by their ability to efficiently manage vast amounts of data generated from a multitude of sources—ranging from traffic sensors to healthcare records—aiming to improve sectors such as energy, healthcare, and community governance, as Figure 1 shows. Despite the advantages, the challenge of data acquisition persists, exacerbated by strict data protection regulations and the growing demand for privacy, which contribute to the formation of fragmented data ecosystems or 'data islands' within urban settings. In response, federated learning emerges as an effective approach to navigate these challenges. This method allows for the decentralized training of models on local data held by various stakeholders, thereby adhering to privacy concerns without centralizing sensitive information. Since its initial introduction by Google [3], the application of federated learning has expanded, driven by ongoing research aimed at enhancing its efficiency and accuracy [4–7]. However, the implementation of federated learning within smart cities is fraught with obstacles, such as high communication costs; difficulties in achieving model convergence in diverse, non-IID

data environments; and the critical need for robust security measures to safeguard against potential data breaches during the model training process [8–11].

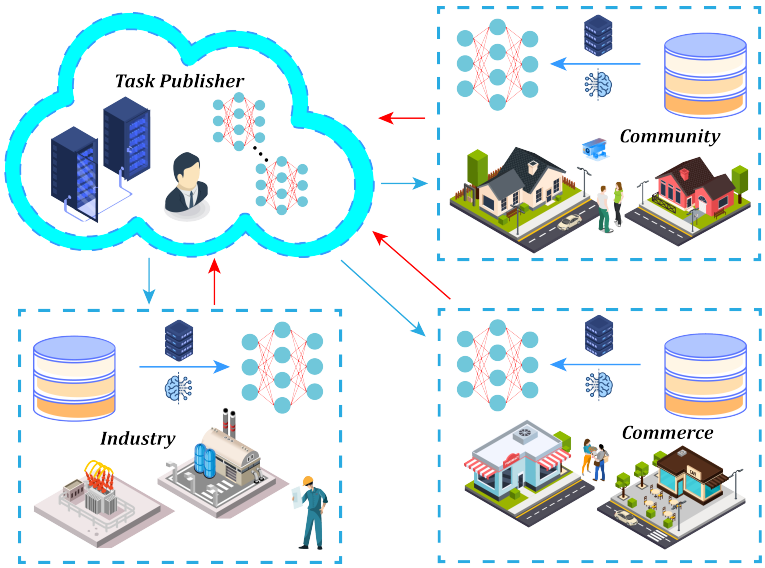

Figure 1. Our application of access control in a smart city.

In existing federated learning systems, the number of clients involved in each round of updates is usually fixed. In the context of a smart city, federated learning schemes normally select a small number of clients randomly to participate in each round, due to the limitations of participants' state and network conditions. However, as there is a mass of heterogeneous clients in reality, such random selection of clients will increase the adverse impact of data heterogeneity [12]. Therefore, it is very important to select appropriate clients for training. Current schemes either select clients with higher statistical utility based on the measurement of their contributions to model updates [13] or select clients based on computing resources and communication constraints [14]. Although these schemes achieve certain effects, there still exist some challenges. For example, some schemes need to analyze private gradients uploaded by participants, or they consume a lot of resources for learning and testing, while some can only select participants at a coarse-grained level.

Federated learning prevents direct uploads of private data, but the issue of privacy leakage has not been completely resolved. Traditional client selection schemes in federated learning typically allow participants to train models with local datasets and upload gradients to update the global model, so that the central server can use this information to avoid model poisoning and select participants for the next round of training to favor model convergence [15,16]. However, some scholars have pointed out that this will also cause serious privacy disclosures [8]. To solve this problem, some studies have used homomorphic encryption [17] and differential privacy [18] to mask the gradient, but this undoubtedly prevents the central server from selecting participants, because the server cannot obtain valid information from the encrypted or confusing gradient. In addition, existing federated learning schemes usually assume that the participants unconditionally use local resources to train the models and upload gradients to the central server, which is not sustainable in reality [19]. Some scholars have looked at federated learning from the perspective of crowdsourcing [20]. Inspired by this, we believe that, in smart cities, the publisher of a federated learning task should have no control over the participants, and the clients should choose whether or not to use local data for training. Therefore, it is necessary to set up an incentive mechanism to attract participants to join the training [11].

In the context of smart cities, we have sufficient reasons to design a federated learning framework from the perspective of crowdsourcing. This framework should consider selecting participants during training to improve the training efficiency, blocking malicious adversaries before training, and encouraging more high-quality clients to participate in con-

structing the models. In recent years, attribute-based encryption has been widely studied as a promising direction of functional encryption [21]. Ciphertext-policy attribute-based encryption (CP-ABE) can conduct fine-grained access control for users conforming to specific policies without revealing any private data. This enables us to separate the participant selection module from the federated learning module, thus providing the possibility of complete privacy protection, including homomorphic encryption. It is worth noting that there is no research on its application in federated learning. In addition, a consortium blockchain is a tamper-resistant and traceable distributed ledger that can be used to record the contributions of participants.

To better understand our scheme, let us consider a scenario in which a company needs to train a model of people's desire to consume different goods. It is hoped that as many clients as possible in the region will participate, even if this is done at a cost. At the same time, the company wishes to eliminate malicious attacks from competitors and select participants with an appropriate data distribution in training to improve the learning efficiency. Although stringent data confidentiality regulations prevent it from deducing the appropriateness from gradients, it can still apply an attribute-based encryption scheme to select participants. Specifically, the task publisher develops a policy for each round of training so that only those who meet this policy can decrypt and participate in subsequent training. At the same time, participants can record decryption logs in a blockchain, which can provide both non-repudiation credentials to incentivize the participants and an auditing report to trace the transactions if a malicious adversary tries to disrupt the model.

The contributions of this article are as follows.

- We propose a client selecting framework in federated learning based on ciphertext-policy attribute-based encryption, which extends traditional federated learning from the perspective of crowdsourcing. Our scheme can select appropriate participants on the premise of protecting gradient privacy.
- An incentive mechanism based on blockchain is proposed, so that the profits to participate in training belong to clients. The use of immutable smart contracts can greatly improve the enthusiasm of clients participating in federated learning.
- The security of the proposed scheme is proven, and the performance of the proposed scheme is evaluated. The experiments show that the method proposed in this paper can perform better than the existing methods.

The rest of our article is organized as follows. Section 2 presents an analysis of related work. Section 3 briefly describes the preliminaries, including the security model of this scheme. Section 4 describes the workflow and the architecture of the proposed CP-ABE scheme. Section 5 characterizes the IND-CPA security model and describes other security proofs. Section 6 compares the performance of our proposed scheme with that of other recent schemes. Finally, Section 7 draws the conclusions.

2. Related Work

The concept of federated learning was proposed by researchers at Google [3], who devised an interesting virtual keyboard application. Federated learning, as defined by Kairouz et al. [9], is a machine learning setting where multiple entities (clients) collaborate in solving machine learning problems, under the coordination of a central server or service provider. Each client's raw data are stored locally and not exchanged or transferred. A typical federated learning process consists of five steps: client selection, broadcast, client computation, aggregation, and model updates. Among them, it is a very challenging task to select appropriate clients during training, rather than performing random selection, and there are still some problems to be solved in the existing client selection schemes.

Zhang et al. [14] selected the clients according to the resource information sent by them, such as the computing ability and channel state. However, this may mean that clients with a large amount of data are unlikely to participate in training. Chai et al. [12] stratified the clients and adaptively selected those with similar training performance per round in order to mitigate heterogeneity without compromising the model accuracy, but this means

that the central server has to control all participants to capture the training time on-the-fly. Fan et al. [22] used importance sampling to select clients, i.e., to select clients by utility. In addition, they developed an exploration–exploitation strategy to select participants. However, each of these clients was designed to upload complete model updates to the central server at each round, ignoring the fact that not all model updates contribute equally to the global model. As an improvement on this work, Li et al. [23] proposed PyramidFL, which calculated the importance ranking of each client based on feedback from past training rounds to determine a list of qualified clients for the next round of training, but the central server still obtains private information, such as the gradients and loss uploaded by clients. Wang et al. [24] put forward an experience-driven federated learning framework (Favor) based on reinforcement learning, which can intelligently select the clients participating in each round of federated learning to offset the deviation caused by non-IID. However, the disadvantage is that the efficiency of reinforcement learning restricts the performance of the system, and sometimes it is unclear why it is effective.

We can consider federated learning from the perspective of crowdsourcing, which may be an important direction for future federated learning because few companies have as many registered users as Google. Thus, we have a strong motivation to respect participants' willingness to participate in training while fully protecting their data. The additional challenge that needs to be addressed to apply federated learning in smart city scenarios is participant motivation [11], and most existing federated learning schemes assume that the participants use local data for training and upload model updates unconditionally. This is not realistic, as participants have the right to claim remuneration for the resources that they consume to participate in training. In order to provide appropriate incentives, Sarikaya et al. [25] designed a Stackelberg game to motivate participants to allocate more computing resources. Richardson et al. [26] designed payment structures based on the impact characteristics of data points on the model loss function to motivate clients to provide high-quality data as soon as possible. In many applications, blockchain is considered to be the best solution to achieve an incentive mechanism, because it is immutable and auditable and has inherent consensus [27]. Almutairi et al. [28] proposed a solution integrating federated learning with a lightweight blockchain, enhancing the performance and reducing the gas consumption while maintaining security against data leaks. Weng et al. [29] proposed a value-driven incentive mechanism based on blockchain to force participants to behave correctly. Bao et al. [30] designed a blockchain platform that allows honest trainers to earn a fair share of profits from trained models based on their contributions, while malicious parties can be promptly detected and severely punished. Most of these blockchain platforms complete the verification and audit of gradient updates via the blockchain itself, while ignoring the costs. Moreover, these pure blockchains overemphasize transactions, without taking into account the difference in data value between different participants. We believe that, from the perspective of crowdsourcing, it is natural for the task publisher to pay high-value participants who meet his/her requirements.

In order to achieve a balance between privacy, performance, and incentives in federated learning, we introduce attribute-based encryption based on ciphertext-policy in participant selection. Sahai and Waters et al. [31] proposed an attribute-based encryption scheme in 2005. Their scheme used a single threshold access structure, and only when the number of attributes owned by users is greater than or equal to a threshold value in the access policy can the ciphertext data be decrypted successfully. Bethencourt et al. [32] first proposed an attribute-based encryption scheme based on ciphertext-policy in 2007. The keys were associated with an attribute set, and the access structure was embedded in the ciphertext. Only when a user's own attribute set meets the access structure set by the data owner can the user successfully decrypt the ciphertext to obtain the ciphertext data, and the access tree structure is used in this scheme. In order to reduce the storage and transmission overhead of the CP-ABE scheme, Emura et al. [33] proposed a scheme with a fixed ciphertext length for the first time, which improved the efficiency of encryption and decryption. However, all these schemes adopt a simple "AND" gate access structure.

Waters et al. [34] proposed a new linear secret shared scheme (LSSS) to represent the access structure, which can realize any monotonous access structure, such as "AND", "OR", and the threshold operation of attributes. This scheme is more expressive, flexible, and efficient.

In smart city scenarios, there are many complex situations, such as the attributes of the participants being revoked. Updating participants' attributes timely and effectively guarantees system security. Pirretti et al. [35] proposed a CP-ABE scheme of indirect attribute revocation in order to solve the loose coupling problem in social networks. Zhang et al. [36] proposed a CP-ABE scheme based on an "AND" gate structure with attribute revocation, but this scheme has poor access structure expression abilities. Hur et al. [37] proposed an access control scheme with coercive revocation capabilities to solve a problem in the access permissions caused by changes in the users' identity in the system. They introduced the concept of attribute groups. Users with the same attributes belong to the same attribute group and are assigned to the same attribute group key. Once a member of the attribute group is revoked, a new group key is generated and sent to all group members except the revoked user. The ciphertext is updated in the cloud with the new group key, which makes it impossible for the revoked user to decrypt the data. However, their scheme does not prevent a collusion attack between the current and revoked users. In order to prevent cooperative decryption between users who have revoked attributes and users who do not have attributes, Li et al. [38] proposed a CP-ABE scheme to resist collusion attacks and support attribute revocation. However, the computational complexity of their scheme is still too high.

To address the challenges identified in the related work, our study introduces a novel federated learning framework that utilizes ciphertext-policy attribute-based encryption (CP-ABE) and a consortium blockchain. This methodology combines the strengths of CP-ABE to provide fine-grained access control and ensure privacy with the transparency and traceability of blockchain to manage and audit participant contributions effectively. The selection of participants based on attribute encryption ensures that only those who meet pre-defined criteria can access and process the training data, thereby enhancing the privacy and security of the data used in our federated learning model. Additionally, the consortium blockchain serves as a decentralized ledger to record all participant activities, which supports non-repudiation and helps in maintaining a trustworthy environment for all parties involved.

3. Preliminary

3.1. Federated Learning

Federated learning is a promising research area for distributed machine learning that protects privacy. In the process of federated learning, the task publisher can train models with the help of other participants. Instead of uploading private data to the central server, participants obtain a shared global model from the server and train it on a local dataset. These participants then upload the gradients or weights of the local model to the task publisher to update the global model. In particular, taking FedAVG as an example, the objective function under federated learning is rewritten with the non-convex loss function of a typical neural network.

$$f(w) = \sum_{k=1}^{K} \frac{n_k}{n} F_k(w) \quad \text{where} \quad F_k(w) = \frac{1}{n_k} \sum_{i \in \mathcal{P}_k} f_i(w)$$

Here, k represents a total of k participants, and n_k represents the number of training set samples in the *k-th* participant. The specific algorithm is quite simple. Firstly, we select some nodes in each batch for epoch training, and then each node uploads weight updates to the server.

$$w \leftarrow w - \eta \nabla \ell(w; b)$$

Then, the server collects all the w_{t+1}^k to obtain the weighted average value of the new global w_{t+1}, and it is then sent to each participant.

$$w_{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} w_{t+1}^k$$

Finally, each participant replaces the w_{t+1} calculated from the last epoch with the delivered update to train a new epoch. The system repeats the above three steps until the server determines w convergence.

3.2. Bilinear Pairing

Bilinear pairing, also known as bilinear mapping, was initiated to build functional encryption schemes. At present, most ABE schemes [39] are based on bilinear pairing cryptography, and its security has been recognized by many experts. The general definition of bilinear pairing is given below.

Consider three cyclic groups G_1, G_2, and G_T, each of prime order p. Typically, G_1 and G_2 are groups of points on an elliptic curve over a finite field, and G_T is a multiplicative group of a finite field. A bilinear pairing is a map

$$e : G_1 \times G_2 \rightarrow G_T$$

that satisfies the following properties.

1. Bilinearity: For all elements $u, v \in G_1$ and $w, z \in G_2$, the pairing operation respects the distributive property over the group operation. That is,

$$e(u \cdot v, w) = e(u, w) \cdot e(v, w)$$

$$e(u, w \cdot z) = e(u, w) \cdot e(u, z)$$

   This property can be extended to the exponents in the groups

$$e(u^a, w^b) = e(u, w)^{ab}$$

   for all $a, b \in \mathbb{Z}$. This property is fundamental in enabling many cryptographic protocols because it allows the pairing operation to "interact" with the group operations in a predictable way.
2. Non-degeneracy: The pairing is non-trivial in the sense that there exists at least some $u \in G_1$ and $w \in G_2$ such that $e(u, w) \neq 1$ in G_T. This ensures that the pairing map is not constantly zero and thus is useful for cryptographic applications. It is often required that for all $u \neq 1$ in G_1 and all $w \neq 1$ in G_2, $e(u, w) \neq 1$.
3. Symmetry (in some cases): For some pairings, particularly symmetric pairings, $G_1 = G_2$ and the pairing satisfies $e(u, w) = e(w, u)$. This symmetry is not always required or desired, depending on the cryptographic application.
4. Computability: There must be an efficient algorithm to compute $e(u, w)$ for all $u \in G_1$ and $w \in G_2$. The efficiency of this computation is critical because the practicality of cryptographic protocols based on pairings depends heavily on the ability to compute these pairings quickly.

Bilinear pairings are not only theoretical constructs but are practically implemented using specific types of elliptic curves, such as supersingular curves or curves with a low embedding degree, which provide the necessary mathematical structure to support efficient and secure pairings. These properties make bilinear pairings powerful tools in modern cryptographic systems, providing functionalities that are not feasible with traditional cryptographic primitives.

3.3. Consortium Blockchain

Blockchain is essentially a decentralized database. It adopts distributed accounting and relies on ingenious algorithms based on cryptography to achieve the characteristics of tamper-proofing and traceability. These features can establish a foundation of trust for a fair distribution of incentives in federated learning [10].

There are three main types of blockchain, namely public chain, private chain, and consortium chain. The essential differences between them are related to who has the write permission and how distributed they are. The public chain is highly decentralized, so anyone can access and view other nodes, but the cost is that the ledgers are very slow to update. At the other extreme is the private chain, where accessing and authoring are entirely controlled by an agency, but this also leads to the excessive concentration of power. The most appropriate blockchain applied in federated learning is the consortium chain, which is jointly maintained by the members and is highly suitable for transaction clearing within the consortium. It is more reliable than the purely private chain and has better performance than the public chain.

Regardless of the type of blockchain applied in a specific scenario, the data structure is a linked list of ledgers containing transaction records, as Figure 2 shows. Each block in the linked list contains hash values of the previous block, a new transaction record, and other information, such as timestamps. This structure ensures that each block is not tampered with and any nodes can easily trace back each transaction along the pointer.

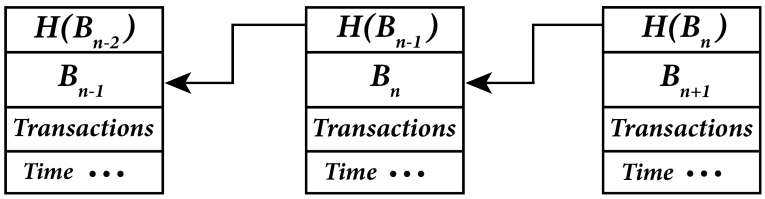

Figure 2. Typical blockchain data structure with hash pointers.

3.4. Security Model

Let $\Pi(Setup, KeyGen, Encrypt, Re-encrypt, Decrypt)$ be our scheme. To define a selective IND-CPA security model for Π, the following $Game_{\Pi,\mathcal{A}}$ game is designed, involving a PPT attacker $\mathcal{A}$ and a PPT challenger $\mathcal{C}$.

Init: An adversary $\mathcal{A}$ controls a series of attribute authorities $AA_k \in AA$ (where at least two authorities in AA are not controlled by $\mathcal{A}$) and the remaining AA are controlled by the challenger $\mathcal{C}$. An adversary $\mathcal{A}$ submits the access structure $\mathbb{A}^*$ to be challenged, and then sends it to challenger $\mathcal{C}$.

Setup: $\mathcal{C}$ runs a setup algorithm in order to obtain the master keys MSK and public parameters PP. Subsequently, challenger $\mathcal{C}$ sends the public parameters PP to adversary $\mathcal{A}$. Meanwhile, challenger $\mathcal{C}$ initializes the user list, which includes authorization attributes and the challenged access structure $\mathbb{A}^*$.

Phase 1: $\mathcal{A}$ adaptively sends a set of attributes S. $\mathcal{C}$ generates the corresponding $SK_1, \ldots, SK_{q_1}$, which is returned to $\mathcal{A}$.

Challenge: $\mathcal{A}$ submits two messages M_0 and M_1 with equal length and submits an access structure $\mathbb{A}^*$ to $\mathcal{C}$. It is required that, for every S queried by $\mathcal{A}$, S cannot satisfy $\mathbb{A}^*$. $\mathcal{C}$ flips a coin $b \in \{0,1\}$ and encrypts M_b with the access structure $\mathbb{A}^*$ to obtain CT^*. Finally, $\mathcal{C}$ sends the ciphertext CT^* to $\mathcal{A}$.

Phase 2: Repeat Phase 1. For every S queried by $\mathcal{A}$, S cannot satisfy the access structure $\mathbb{A}^*$.

Guess: $\mathcal{A}$ outputs a guess $b' \in \{0,1\}$ for b and wins the game if $b' = b$.

The advantage of $\mathcal{A}$ is defined in this game as follows:

$$Adv(\mathcal{A}) = \left| \Pr[b' = b] - \frac{1}{2} \right|. \quad (1)$$

We note that the model can easily be extended to handle chosen-ciphertext attacks by allowing for decryption queries in Phase 1 and Phase 2.

Definition 1. *The protocol* Π *is CPA security if no probabilistic polynomial-time (PPT) adversaries have a non-negligible advantage in the above game.*

Under our security model, the task publisher and its central servers are considered to be honest but curious. In other words, they do not counterfeit, attack, or try to decipher the data uploaded by the owners, and they faithfully execute the algorithms. However, they may have a certain degree of curiosity and may bypass some restrictions to access users' data or the system parameters directly. Meanwhile, the participants may be malicious, and they may attempt to access data that exceed their permissions in collusion with others.

4. Proposed Scheme

In this section, we provide our proposed system framework and details of our scheme, and we then verify their appropriateness. Figure 3 shows the framework diagram of our scheme.

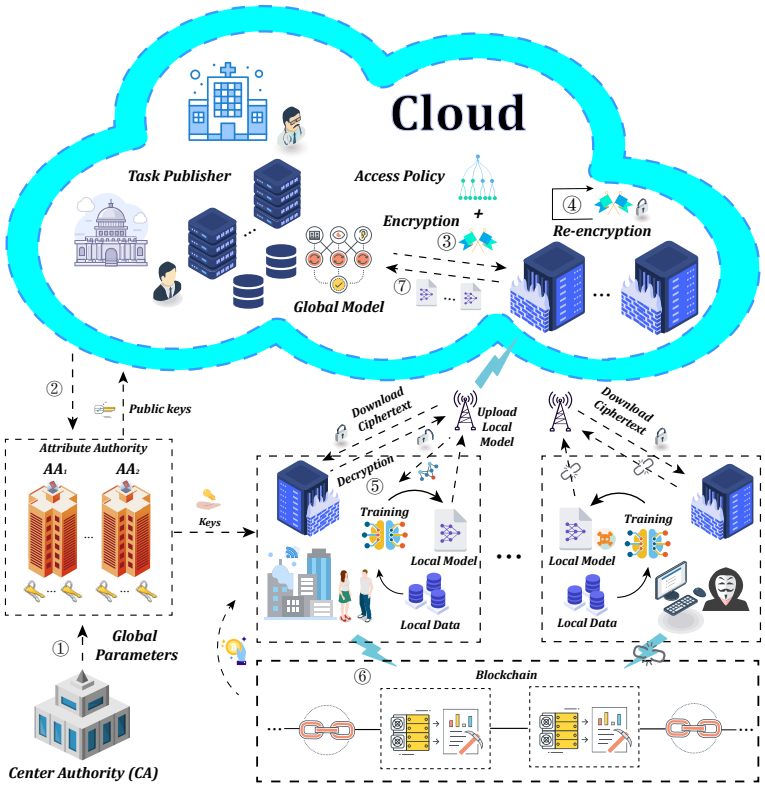

Figure 3. Framework of the proposed scheme.

4.1. System Framework

① The central authority (CA) receives a security parameter λ and generates public parameters (PP) before publishing them in the system.

② The task publisher tries to build a global model by selecting a set of attribute names and delegating attribute authorities to generate different attribute value keys for potential participants.

③ The task publisher initializes the model weights and establishes an access policy before generating a linear secret sharing matrix (M, ρ). Then, he/she uses public keys obtained from AAs to encrypt a flag as a credential to participate in the current communication round.

④ If some participants' attributes change, the task publisher obtains the latest version of the attribute public keys from the attribute authorities, re-encrypts the ciphertext, and attaches a digital signature.

⑤ Participants download the ciphertext from the central server (CS), verify the signature, and then perform decryption operations. If a participant meets the access policy set by the task publisher, such as requirements on the data quantity, data quality, and computing ability, he can successfully decrypt the ciphertext and obtain the flag. After an interaction with the server, such as homomorphic encryption key negotiation, the participant can use local data to carry out the next round of updates and return the updated weights to the central server.

⑥ Participants upload the decrypted flag of the current round to the consortium chain as a credential for an incentive.

⑦ After verifying the flag sent by the selected participants, the publisher can use the weight update to calculate new global weights and repeats this process until the global model converges.

4.2. Algorithms

We describe the specific algorithm as follows.

4.2.1. Global Setup: $\boldsymbol{Setup(\lambda) \rightarrow PP}$

The central authority(CA) firstly selects a system security parameter λ, and then selects a large prime p as the order of multiplicative groups $\mathbb{G}$ and $\mathbb{G}_{\mathbb{T}}$. Thus, $e : \mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_{\mathbb{T}}$ is the bilinear map. Let g be the generator of $\mathbb{G}$. Finally, it chooses a hash function $H : \{0,1\}^* \rightarrow \mathbb{G}$, used to map binary sequences such as identifiers or attribute values to elements in a group.

After these top-level parameters are set, the central authority runs the initialization algorithm. It generates the master keys MSK by choosing $\alpha, \tau, a \in \mathbb{Z}_p$ randomly.

$$MSK = (\alpha, \tau, a)$$

Then, the public parameter PP is as follows:

$$PP = (g, g^a, e(g,g)^\alpha)$$

In addition, all AAs are required to register with the CA to obtain unique identifiers *aid* that are used to prove their legal identities.

4.2.2. Key Generation: $\boldsymbol{KeyGen(GP) \rightarrow PK}$

In our system, each AA manages different attributes. The attribute authority with an identifier *aid* is denoted as AA_{aid}, and the attribute set managed by is S_{aid}. Once an AA is initialized, it begins to execute a series of key generation programs.

When a task publisher needs to publish a federated learning task, he can pre-determine the attributes of the participants involved in the training, such as the computational performance, data set distribution, data quantity, and the willingness to participate, etc. He then instructs the attribute authority to generate the associated attribute keys on his behalf.

First, to distinguish between different versions of the attribute keys due to attribute revocation, the authority chooses a random number $v_x \in \mathbb{Z}_p$ as the initial version number of attribute x. The public attribute keys PK_x are generated as

$$PK_x = \{PK_{1,x} = H(x)^{v_x}, PK_{2,x} = H(x)^{v_x \tau}\}$$

In particular, the attribute authority is also responsible for updating the keys. If the attribute x of a participant changes, the authority runs the algorithm $NKeyGen(MSK, VK)$ to generate update keys. The inputs are the new version number v_x^n corresponding to

attribute x and the master key MSK. It generates the current attribute keys by choosing a number v_x^n randomly. After this, the authority computes the update keys as

$$UK_x = \left\{ UK_{1,x} = \frac{v_x^n}{v_x}, UK_{2,x} = \frac{v_x - v_x^n}{v_x \tau} \right\}$$

The new version number of the attribute x, the update keys UK_x that can be used to update the secret keys of unrevoked participants, and the ciphertexts that are associated with the revoked attribute x are the outputs of this algorithm. In addition to generating the update keys, the attribute authority also needs to update the public keys of the revoked attribute as

$$PK_x^N = \left(PK_{1,x}^N = H(x)^{v_x^n}, PK_{2,x}^N = H(x)^{v_x^n \tau} \right)$$

The attribute authority then sends the update keys to the task publisher and all parties that have not been revoked via a corresponding attribute over a secure channel. The new public attribute keys for the revoked attribute are available to all owners from the institution's public bulletin board. All generated secret keys are centrally managed by AA and isolated from outside. Malicious adversaries cannot obtain any information about the private keys through the network, but all public keys are publicized.

4.2.3. Registration: $\boldsymbol{UserReg(MSK, VK, S) \rightarrow SK}$

Participants in federated learning can be community residents with valuable data. From the perspective of crowdsourcing, when an institution publishes a federated learning task, since each participant has absolute control over their own data, they can independently decide whether to join the training of this model and obtain certain benefits. On the other hand, each participant has a high degree of specificity, as their computing power, the amount of data, and the data distribution is different. Therefore, they need to register with the trusted attribute authority before participating in the training, so as to obtain the corresponding attribute keys according to their respective computed value and data value.

When a participant attempts to join the federated learning system, he can declare his set of attributes to the attribute authority for verification. If the information provided by the client is sufficient to prove the set of attributes that he claims, the attribute authority runs the key generation algorithm $UserReg$ to generate the unique secret keys SK for the participant. The algorithm takes the master keys MSK, a set of attributes S, and the version number $\{v_x\}_{x \in S}$ corresponding to the attributes as inputs. It then computes the participants' secret keys by choosing a random number $t \in \mathbb{Z}_p$ as

$$SK = \{ K = g^{\alpha} \cdot g^{at}, L = g^t, \forall x \in S : K_x = H(x)^{v_x t} \}$$

When a certain attribute x of a user is revoked—for example, he leaves an organization—the attribute authority needs to update the decryption private keys for other members of the attribute group, as follows:

$$K_x^N = K_x^{UK_{1,x}} = H(x)^{v_x^n t}$$

The attribute authority returns the updated keys over a secure channel to the users who have not been revoked. If the attribute of a participant is revoked, the participant cannot use the previous attribute keys to decrypt the ciphertext. However, a participant whose attributes are not revoked only needs to update the keys corresponding to the revoked attribute as

$$SK^N = \left(K, L, K_x^N, \forall x' \in S \backslash \{x\} : K_{x'} \right)$$

4.2.4. Server Encryption: $\boldsymbol{Enc(GP, PK, f, \mathbb{A}) \rightarrow CT}$

When a task publisher needs to share global gradient information updated in each training round with the participants, firstly, the task publisher needs to formulate an appropriate access policy according to the model to be trained. The specific principle is that no information, including gradients, can be obtained from a single participant, and all participants who meet the access policy can obtain positive benefits after model training. For example, the task publisher can specify quantitative indicators to compute the ability, data quantity, degree of independence, and data distribution.

The algorithm takes in the access policy created by the task publisher and then outputs an $n \times l$ LSSS access matrix M with $\rho(x)$ mapping its rows to attributes. Now, $\mathbb{A} = (M, \rho)$, where $\rho = (att_{\rho(1)}, att_{\rho(2)}, \ldots, att_{\rho(n)})$.

Typically, in order to fully ensure gradient privacy, the participant may use homomorphic encryption with the server to protect the updated gradient information, which requires the negotiation of the homomorphic encryption keys with the task publisher. This means that, before each round of training, the task publisher needs to identify who the participants are. To do this, the central server secretly selects a flag f as a credential to participate in the training round. Participants who can successfully decrypt and return the f can participate in the next training round. Therefore, the flag serves as the ciphertext that needs to be encrypted.

After this, the central server (CS) chooses a random vector $\xi \in \mathbb{Z}_p^l$ with s as its first entry. Let λ_i denote $M_i \cdot \xi$, where M_i is the row i of M. For each $i \in [1, n]$, the central server randomly chooses $r_i \in Z_p$ and computes the following ciphertext:

$$\begin{aligned} &C = fe(g,g)^{\alpha s}, C' = g^s, \\ &C_{0,i} = g^{a\lambda_i} H(\rho(i))^{-r_i v_{\rho(i)}}, \\ &C_{1,i} = H(\rho(i))^{v_{\rho(i)} r_i \tau}, \\ &C_{2,i} = g^{r_i} (i = 0, \cdots, n-1) \end{aligned}$$

Lastly, CS generates ciphertext CT.

$$\begin{aligned} CT = \{&(M, \rho(i)), C, C', \\ &C_{0,i}, C_{1,i}, C_{2,i} | i \in [1, n]\} \end{aligned} \tag{2}$$

As is well known, the attributes owned by participants in federated learning may change dynamically over time. Thus, in order to support attribute revocation, the central server controlled by the task publisher needs to re-encrypt the ciphertext. In other words, when a participant's attributes are revoked, the central server re-encrypts the ciphertext to prevent malicious or inappropriate participants from training the model. If some user's attribute x' is revoked, the central server receives an updated message sent by some of the attribute authorities. Assume that the updated key is UK_x. After re-encryption, the new ciphertext is as follows:

$$\begin{aligned} CT^N = \Big(&C^N = C, C'^N = C', \forall i = 0 \text{ to } n-1 : C_{2,i}^N = C_{2,i}, \\ &\rho(i) \neq x' : C_{0,i}^N = C_{0,i}, C_{1,i}^N = C_{1,i} \\ &\rho(i) = x' : C_{0,i}^N = C_{0,i} \cdot (C_{1,i})^{UK_{2,x'}}, C_{1,i}^N = (C_{1,i})^{UK_{1,x'}} \Big) \end{aligned}$$

Finally, to achieve IND-CCA security, the central server runs a signature algorithm to obtain verification key vk and signing key sk, after which the cloud runs $Sign_{sk}(CT) \rightarrow \sigma$. Note that an adversary cannot forge a new signature on a message that has been signed previously.

$$Final\ ciphertext = (vk, CT, \sigma) \tag{3}$$

It is worth mentioning that because homomorphic encryption is used to completely protect the privacy of a participant's upload gradient, the task publisher cannot access the participant's data. Therefore, the selection of suitable participants by the central server is based on the authentication of the flag. When a participant decrypts and obtains the flag successfully within the deadline, the task publisher can include it in the node pool of this round of training. The central server can then negotiate homomorphic encryption keys with these participants and execute federated learning algorithms, such as Fedavg.

4.2.5. Participant Decryption: $\boldsymbol{Dec(CT, SK) \rightarrow flag}$

Firstly, a potential participant obtains a ciphertext from the central server and checks whether $\mathrm{Ver}_{vk}(CT;\sigma) \stackrel{?}{=} 1$. If it does not hold, the client outputs $\perp$. Otherwise, it proceeds.

After successful verification, it selects an appropriate $\omega_i \in \mathbb{Z}_p$ with polynomial time complexity, to make $\sum_{P(x)\in S'} \omega_i M_i = (1, 0, \cdots, 0), i \in [1, n]$ true. If it can find such a set of constants $\{\omega_i\}$, the decryption algorithm continues to execute as $s = \sum_{i\in I} \omega_i \lambda_i$; otherwise, it terminates and outputs $\perp$.

The decryption algorithm first computes as follows:

$$
\begin{aligned}
&\frac{e(C', K)}{\prod_{i\in I}\left(e(C_i, L)e\left(K_{\rho(i)}, D_{2,i}\right)\right)^{w_i}} \\
=&\frac{e\left(g^s, g^\alpha \cdot g^{at}\right)}{\prod_{i\in I}\left(e\left(g^{a\lambda_i}H(\rho(i))^{-r_i v_{\rho(i)}}, g^t\right)e\left(H(x)^{v_{\rho(i)}t}, g^{r_i}\right)\right)^{\omega_i}} \\
=&\frac{e(g,g)^{\alpha s}e(g,g)^{ast}}{\prod_{i\in I}\left(e\left(g^{a\lambda_i}, g^t\right)e\left(H(\rho(i))^{-r_i v_{\rho(i)}}, g^t\right)e\left(H(x)^{v_{\rho(i)}t}, g^{r_i}\right)\right)^{\omega_i}} \\
=&\frac{e(g,g)^{\alpha s}e(g,g)^{ast}}{\prod_{i\in I}\left(e(g,g)^{at\lambda_i}e(H(\rho(i)), g)^{-r_i v_{\rho(i)}t}e(g, H(x))^{r_i v_{\rho(i)}t}\right)^{\omega_i}} \\
=&\frac{e(g,g)^{\alpha s}e(g,g)^{ast}}{\prod_{i\in I}\left(e(g,g)^{at\lambda_i\omega_i}\right)} \\
=&\frac{e(g,g)^{\alpha s}e(g,g)^{ast}}{e(g,g)^{ast}} \\
=&e(g,g)^{\alpha s}
\end{aligned}
$$

Then, it can decrypt the flag as

$$f = \frac{C}{e(g,g)^{\alpha s}}$$

Upon acquiring the flag, the participant can send it to the central server to indicate that it meets the policy set by the task publisher and can participate in this round of training, without compromising their privacy. The complete algorithm is shown in Algorithm 1. At the same time, the flag is uploaded to the blockchain to receive the revenue after the training is done.

Algorithm 1 FedAvg-ABE. The K clients are indexed by k; B is the local minibatch size; E is the number of local epochs; and η is the learning rate.

Server executes:
initialize w_0
for each round $t = 1, 2, \dots$ **do**
  $m \leftarrow max(C \cdot K, 1)$;
  $Enc(GP, PK, f, \mathbb{A}) \rightarrow CT$
  for each appropriate client **in parallel do**
    $[[w_{t+1}^k]] \leftarrow$ ClientUpdate(k, CT)
  end for
  $[[w_{t+1}]] \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} [[w_{t+1}^k]]$
  $w_{t+1} \leftarrow [[w_{t+1}]]$ // homomorphic decryption
end for

ClientUpdate(k, CT**):** // Run on client k
if Not match policy: **then**
  return $\perp$
else
  $Dec(CT, SK) \rightarrow flag$
end if
Negotiates the keys of homomorphic encryption with server
$\mathcal{B} \leftarrow$ (split $\mathcal{P}_k$ into batches of size B)
for each local epoch i from 1 to E **do**
  for batch $b \in \mathcal{B}$ **do**
    $w \leftarrow w - \eta \nabla \ell(w; b)$
    $[[w]] \leftarrow w$ // Multi-key homomorphic encryption
  end for
end for
return $[[w]]$.

4.2.6. Incentive: $\boldsymbol{Inc(CT, CID, Time) \rightarrow (Trans.)}$

Since data and computing resources are valuable, participants that use local data and local computing resources should be paid. In this work, if a participant runs the ABE decryption algorithm and obtains updated global gradient information from the central server, which also means that the participant meets a series of policies formulated by the central server, his local data have been used reasonably. Therefore, the task publisher should pay them after the training according to each participant's contribution to the global model. Specifically, in each round of training, participants decrypt messages to obtain the flag that signifies successful decryption, before using their own digital signature to sign the flag and upload it to the non-repudiation consortium chain, where smart contracts are executed. If the client's *cid* is in the list provided by the central server, indicating that the participant is entitled to the benefits arising from the training round, these records are recorded in the blockchain. After the training, the server can trace back the blockchain and distribute the actual profits to the various participants.

As shown in Algorithms 2 and 3, the incentive mechanism can be divided into an upload transaction and confirm transaction. Before each round of training, the task publisher needs to select a flag as the voucher of this round for profit distribution, and each participant tries to decrypt and obtain the flag. The participant and the task publisher together generate the upload transaction TX_{upload} and send it to the blockchain data pool. The transaction is then broadcast to other nodes in the blockchain for verification. Once the deal is validated, it is packaged into the consensus block via PBFT. At the end of the training, the consortium blockchain can be backtracked and the revenue can be distributed to all clients who participated in the training.

Algorithm 2 Upload Transaction Generation

Input: f, cid, $time$
Output: TX_{upload};

1: The task publisher releases the flag f of round t, computes $\xi = H(f||t)$, and sends it to the blockchain secretly.
2: The participant sends client id cid and $time$ to the blockchain
3: Let f_c be the flag decrypted from the ciphertext
4: Compute

$$\zeta = H(H(f_c||t)||cid||time)$$
$$sign = Sign_{cid}(\zeta)$$

5: **return** $TX_{upload} = \{sign, cid, H(f_c||t), time\}$

Algorithm 3 Confirm Transaction Generation

Input: TX_{upload}
Output: $Succ$ or $Fail$;

1: Blockchain nodes receive transaction TX_{upload};
2: The node calculates

$$\zeta = Verify_{cid}(sign)$$
$$\zeta' = H(H(f_c||t)||cid||time)$$

3: **if** $\zeta = \zeta'$ **then**
4: **if** $\xi = H(f_c||t)$ **then**
5: Execute smart contract to allocate the revenues of round t to participants corresponding to cid.
6: **return** $Succ$
7: **end if**
8: **end if**
9: **return** $Fail$

5. Security Analysis

Before we begin our security analysis, we need to clarify the security assumptions of the various entities in the system. First, attribute authorities are considered to be fully trusted entities, similar to certificate authorities, generally initiated by city governments. The task publisher can be a commercial institution, which is reflected in the system as honest and curious, i.e., they faithfully execute the algorithms that they are responsible for without maliciously destroying the ciphertext uploaded by the clients, but they may spy on or infer the clients' private information from the access record. Finally, there may be malicious clients in the system, trying to collude with other clients to obtain data beyond their own permissions or trying to destabilize the system.

5.1. Selective CPA Security

Theorem 1. *There is no polynomial adversary that can selectively break our system with a challenge matrix of size $l^\star \times n^\star$, where $n^\star \leq q$, when the decisional q-parallel BDHE assumption holds.*

Proof. Inspired by Waters [34], we can build a simulator $\mathbb{B}$ that solves the decisional q-parallel BDHE problem with a non-negligible advantage under the prerequisite that none of the updated secret keys SK^N that are generated by both the queried secret keys SK and update keys UKs can decrypt the challenge ciphertext. This is based on the assumption that we have an adversary $\mathcal{A}$ that chooses a challenge matrix $M^\star$ with the dimension of at most q columns with a non-negligible advantage $\epsilon = Adv_{\mathcal{A}}$ in the selective security game against our construction. The proof is produced by the challenger and the attacker through a series

of interactions in the game. Because the mathematical discussion of the game details is beyond the scope of this article and it resembles Waters' work, it is omitted. □

5.2. Data Security

In our scheme, only users with specific attributes can obtain the corresponding keys through the attribute authorities. Since the underlying protocol is based on elliptic curves, and ECDLP is unsolvable, clients without the correct attributes cannot obtain any information about the private keys from the corresponding public keys in polynomial time.

Based on the training progress and results, the task publisher will select the access policy and the flag f of the training round, which is hidden in ciphertext C. Since s is randomly chosen by the task publisher, it is a random number in the eyes of an attacker. Thus, the attacker cannot obtain any valuable information about f. With a linear secret sharing scheme, s is a secret divided by λ_i and can only be recovered if there are enough parts; in other words, the ciphertext can only be decrypted if the participant has a set of attributes that match the access policy. For any invalid users who do not have the attributes declared by the access policy, since they do not have the attributes corresponding to rows of M, they do not make $\sum_{\rho(i)\in S'} \omega_i M_i = (1, 0, \cdots, 0)$ true, where $\omega_i \in \mathbb{Z}_p$. Then, they cannot compute the first element of ξ, which is s. Therefore, this scheme ensures data security.

5.3. Forward and Backward Security

Forward security means that any clients that have been revoked cannot access subsequent data unless the remaining set of attributes of the client still satisfies the access structure. In the scheme proposed in this paper, if the attributes of a client are revoked, only some of the keys and the ciphertext are updated by the central server, which not only reduces the local computational overhead but also effectively prevents clients who have lost access permissions from posing threats to the updated ciphertext in the system, so as to ensure forward security. Considering that the revoked client already has permission to read the old ciphertext, the central server must restrict him from downloading the old ciphertext.

Backward security means that new clients cannot decrypt previously encrypted data. Note that we use *ver* to control the ciphertext version; thus, new clients cannot decrypt the old ciphertext using the latest version of the attribute keys.

5.4. Collusion Attack

Theorem 2. *The scheme is secure under a multi-user collusive attack.*

Proof. In the proposed scheme, the attribute authority will assign a random value $t \in \mathbb{Z}_p^*$ to each participant. Even if multiple participants have exactly the same attribute, the value will be different in the keys obtained by them. In the decryption algorithm, t must be consistent to realize a collusion attack. Therefore, no client can conspire with other users or groups of users to illegally decrypt the data. For example, one participant P_0 has attributes $\mathcal{A}$, and the other participant P_1 has attributes $\mathcal{B}$; for an access policy of "$\mathcal{A} \cap \mathcal{B}$", individual participants P_0 or P_1 cannot decrypt the data alone. Even if they use their attribute keys with $\mathcal{A}$ and $\mathcal{B}$ to collude, the calculation cannot eliminate t; thus, they are unable to perform decryption. □

Tseng et al. [40] found that some attribute-based encryption (ABE) schemes [41,42] based on elliptic curve scalar multiplication are vulnerable to collusion attacks, because users with the same attributes can obtain the attribute private key set in the system by solving linear equations. Our scheme does not have this problem because we use bilinear pairing instead of scalar multiplication, and no party can obtain the secret parameters of the system by solving the equations.

6. Performance Comparison and Evaluation

In this section, we use public datasets to evaluate the performance of our scheme and compare it with previous work. In particular, in addition to showing how the proposed

scheme improves the model accuracy in federated learning, we analyze the impact of using attribute-based encryption on the computational efficiency.

First, in Table 1, we present the characteristics of the currently popular federated learning client selection schemes. It can be seen that our proposed scheme comprehensively considers the dimensions of the client data quantity, data distribution, and computing power, avoiding complex importance measurements and reinforcement learning. We then qualitatively evaluate our work against some of the known incentive mechanisms. As shown in Table 2, most of the existing schemes use either the quantity or quality of data to distribute revenues fairly. Fortunately, the task publisher in our scheme can consider two aspects comprehensively to formulate an access policy, which is more applicable to reality. With the help of the blockchain, we can easily implement the features of auditing and traceability. This is why we use post-training allocation rather than simultaneous allocation during training, to reduce the cost of evaluating the contributions of each participant.

Table 1. Comparison of client selection schemes.

Schemes	System Heterogeneity	Statistical Heterogeneity	Privacy	Expansibility	Fine-Grained	Main Idea
Nishio 2019 [43]	✓	×	×	✓	×	Select as many clients as possible within a specified deadline
Cho 2020 [44]	✓	✓	×	×	×	Select clients with higher local losses
Chai 2020 [12]	✓	✓	✓	×	×	Select clients with similar response latencies
Lai 2021 [22]	✓	✓	✓	✓	×	Select clients through importance sampling
Zhang 2021 [14]	×	✓	×	×	×	Select clients with lower non-IID degrees of data
Wu 2022 [45]	×	✓	×	×	×	Select clients by comparing the gradients of the local and the global
Li 2022 [23]	✓	✓	✓	×	✓	Select clients with higher importance ranking
Our scheme	✓	✓	✓	✓	✓	Select clients using attribute-based encryption

Table 2. Comparison of incentive mechanisms.

Schemes	Data Quality	Data Quantity	Privacy	Efficiency	Auditability	Universality	Main Idea
Song 2019 [46]	✓	×	×	low	×	×	Measure the contribution with a Contribution Index (CI)
Yu 2020 [47]	✓	×	✓	mid	×	✓	Participants dynamically receive payoff according to contributions
Zeng 2020 [48]	✓	×	✓	high	×	✓	Auction theory
Zhan 2020 [49]	×	✓	✓	low	×	✓	DRL-based reward allocation
Weng 2019 [29]	×	✓	✓	mid	✓	×	Use blockchain to record the process of federated learning
Bao 2020 [30]	×	✓	✓	mid	✓	×	Provide a healthy marketplace for collaborative training models
Our scheme	✓	✓	✓	high	✓	✓	Select clients using attribute-based encryption

Next, we describe some details of the experiments.

6.1. Setup

We trained popular convolutional neural network models on two benchmark datasets, FashionMNIST and CIFAR-10. The convergence speed and the final model accuracy of the proposed ABEFedAvg algorithm are compared with three other federated learning aggregation algorithms FedAvg [3], FedProx [50] and FedIR [51] with randomly selected clients. The specific experimental Settings are as follows:

Hardware and Software setup: This paper conducts experiments on a set of Linux servers, each running one experimental task. After all resources have been allocated, the hardware and software setup of each server is shown in Table 3.

Table 3. Hardware and Software setup.

Hardware and Software	Setup
CPU	Intel® Core™ i9-9900X CPU @ 3.50 GHz
Memory	128 G
GPU	NVIDIA GeForce RTX 2080 Ti × 8
CUDA Version	12.0
Programming Language	python3.9
Operating System	Ubuntu 18.04.6 LTS
Federated Learning Framework	Pytorch 1.10.2

Dataset: We comprehensively evaluate the efficiency of ABEFedAvg in simulation experiments using different datasets, namely FashionMNIST and CIFAR-10, which contain numerous fixed-size images and have been used in a large number of studies. The dataset, validation set and test set are allocated to different parties with different data distribution patterns according to Dirichlet distribution to evaluate the performance of ABEFedAvg under non-independent and identically distributed data. The FashionMNIST dataset is a very classic dataset in the field of machine learning. It consists of 60,000 training samples and 10,000 test samples, each of which is a 28 × 28 pixel image representing an item numbered from 0 to 9. The CIFAR-10 dataset has a total of 60,000 color images, each with a scale of 32 × 32 pixels, and is divided into 10 categories with 6000 images each. Of these, 50,000 images are used for training to form five training batches of 10,000 images each, and the remaining 10,000 images are used for testing to form a separate testing batch.

Party: Then this paper uses the method in [52] to generate the partition of Non-IID. Specifically, the parameters of the Dirichlet distribution are set to partition the dataset to different parties in an unbalanced manner. When the parameter α is larger, the data of each party tends to be independently and identically distributed. On the contrary, the data distribution is more uneven. In this paper, three distribution cases are set, and $\alpha = inf$ is used to simulate the ideal situation where the data is completely independent and identically distributed, as Figure 4 shows. Use $\alpha = 0.5$ to simulate a slightly independent and identically distributed scenario, which is common in real-world scenarios, as Figure 5 shows. We use $\alpha = 0.1$ to simulate a worst-case data distribution where almost each party has only 3–4 classes, as Figure 6 shows. The data distribution of each parameter setting participant is shown as follows:

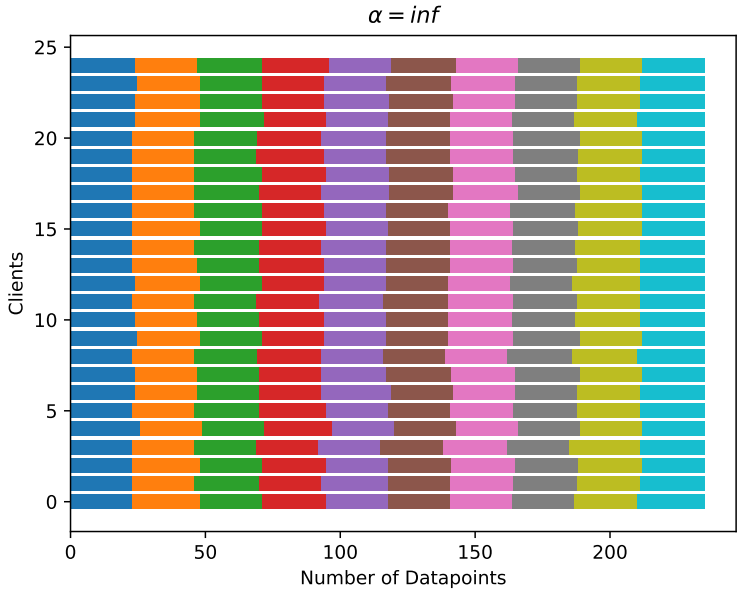

Figure 4. Completely IID.

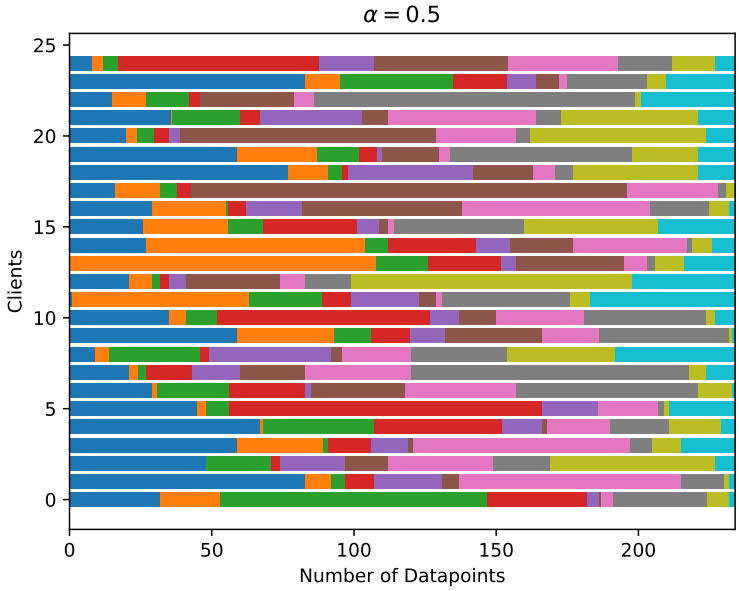

Figure 5. Slightly IID.

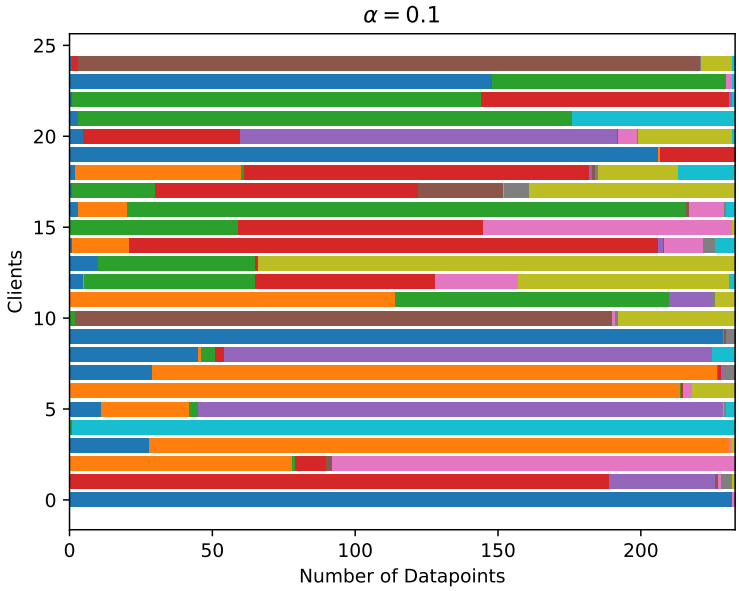

Figure 6. Worst-case.

Model: The model used in this article is LeNet-5 convolutional Neural Network (CNN), which is commonly used for image classification. The model structure of LeNet-5 includes convolutional layer, pooling layer and fully connected layer. The convolutional layer and pooling layer are used to extract the local features of the image, and the fully connected layer is used to map the features to the class probabilities. The first and third layers are convolutional layers with 6 and 16 kernels, respectively, each of size 5×5 and step size 1; Convolutional layers are followed by average pooling layers with a pooling kernel of size 2×2 and no padding is used, their role is to downsample the input feature map and reduce the size of the feature map. The last three layers are fully connected layers with 120, 84 and 10 neurons, respectively. In the convolution kernel and the fully connected layer, ReLU is used as the activation function to avoid the problem of gradient disappearance. For the FashionMNIST dataset, the input image is $28 \times 28 \times 1$, while the CIFAR-10 dataset has an input image specification of $32 \times 32 \times 3$.

Performance index: In order to evaluate the optimization degree of the proposed party selection mechanism based on attribute encryption on various synchronous federated learning algorithms, this paper uses the test set accuracy as the main indicator to measure the performance of the model, trains on the FashionMNIST dataset and CIFAR-10 dataset for 500 rounds and 1000 rounds respectively, and plots the test set accuracy curve. Finally, the average accuracy and the highest accuracy are calculated, where the accuracy is defined as the ratio of the number of correctly classified images to the total number of test sets, and the range is between 0 and 1. To evaluate the convergence speed of the proposed ABEFedAvg algorithm, the number of communication rounds for the model to converge to the target accuracy, ToA@x, is used as the main metric to measure the efficiency of model training, where x represents the target accuracy.

6.2. Experimental Results

6.2.1. Effect of the Number of Participant Selection on Performance

Firstly, we study the impact of using the stringency of the access policy in the proposed attribute-based encryption participant selection scheme and the participant selection score $\mathcal{C}$ of the baseline algorithm FedAvg on the performance of federated learning. In this paper, we assume that there are $K = 100$ parties in a region, and three different access strategies are selected, and the stringency is set to "strict", "moderate" and "loose" respectively. The corresponding comparison of the three participant selection scores is as follows. $mathcalC = 0.1$, $mathcalC = 0.2$, $mathcalC = 0.3$. The performance evaluation of different access strategies and selection scores using FashionMNIST and CIFAR-10 picture datasets is shown in Table 4.

Table 4. Training results for different number of participant selection.

Algorithm	Fraction Size (C)	Average Accuracy		Highest Accuracy		ToA@0.85 ToA@0.7	
		F-MNIST	CIFAR-10	F-MNIST	CIFAR-10	F-MNIST	CIFAR-10
FedAvg	c = 0.1	0.8318	0.6498	0.8567	0.6809	393	-
ABEFedAvg		0.8816	0.7346	0.8863	0.7433	70	241
FedAvg	c = 0.2	0.8631	0.7046	0.8713	0.7121	175	669
ABEFedAvg		0.8943	0.7508	0.8974	0.7583	62	167
FedAvg	c = 0.3	0.8778	0.7115	0.8803	0.7153	127	462
ABEFedAvg		0.8893	0.7378	0.8912	0.7412	73	195

For the FedAvg algorithm, when $\mathcal{C} = 0.3$, the accuracy of the model in the test set is the highest, and with the decrease of the participant selection score, the accuracy also decreases in turn. When $\mathcal{C} = 0.1$, the accuracy is only 0.8318, and the training curve has the largest degree of fluctuation. This is because the small number of selected parties in each round of training reduces the number of samples for model learning. When the selection score $\mathcal{C}$ is 0.3, the model training time is the shortest, only 127 rounds are needed. When the selection score $\mathcal{C}$ is reduced to 0.2, the number of communication rounds increases by 48 rounds. The training time of the model grows substantially, requiring 393 rounds of communication to reach the target accuracy, an increase of 266 rounds compared to the setting with $\mathcal{C} = 0.3$. Therefore, in order to balance the model accuracy and training time overhead, the party selection score is set to $\mathcal{C} = 0.2$ in the following experiments.

For ABEFedAvg algorithm, the key to affect the number of selected parties is the stringency of the access policy. When using the "strict" access policy, the central server only allows the subjects with the largest number of samples and the most uniform distribution of all participants to participate in the training, while using the "loose" access policy means accepting the participants with low degree of independent and identical distribution. Experimental results show that the model has the highest accuracy when selecting the "moderate" access strategy, reaching an average accuracy of 0.8943 and 0.7508 on the

FashionMNIST and CIFAR-10 datasets, which shown in Figures 7 and 8, respectively. This is because choosing a more stringent access policy can improve the quality of the selected parties, but it also rejects more data samples that are still valuable for training. On the contrary, choosing a more relaxed access policy will weaken the effect of access control and introduce more parties with uneven local sample data. Based on this, the "moderate" access policy is selected in the following experiments.

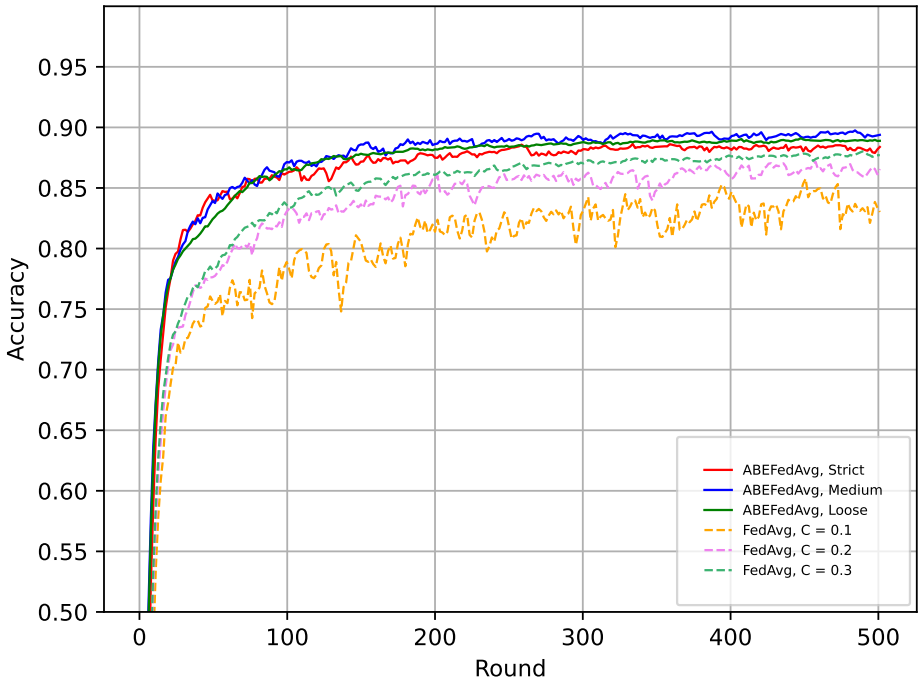

Figure 7. FashionMNIST Test accuracy for different number of participant selection.

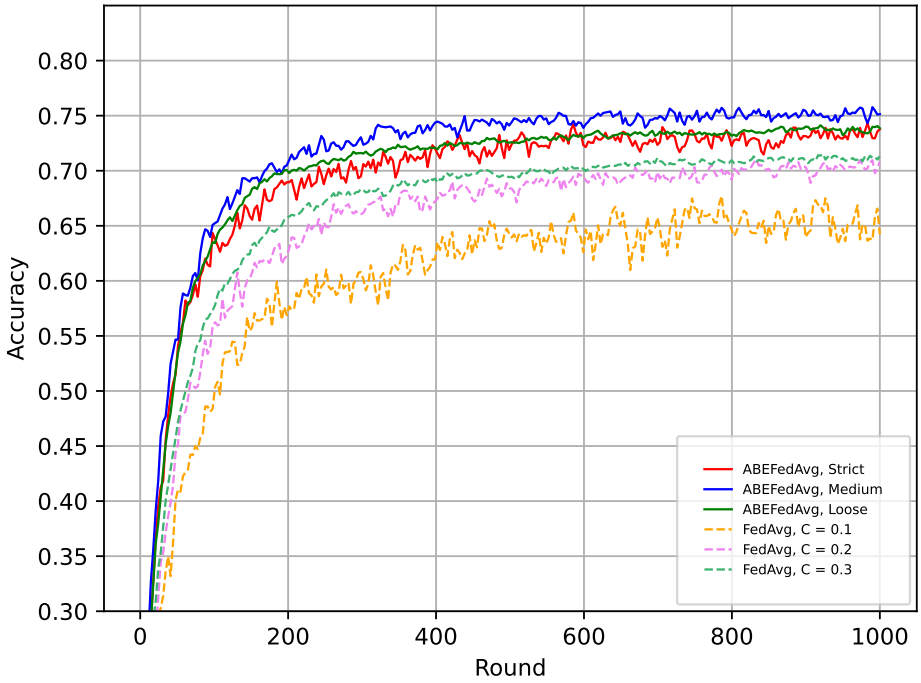

Figure 8. CIFAR-10 Test accuracy for different number of participant selection.

6.2.2. Influence of Independent and Identically Distributed Data on Performance

It is well known that in real scenarios, the degree of independence and identically distributed data of each participant in federated learning is often unpredictable. Generally speaking, the higher the degree of independence and identically distributed data of each participant, the better the accuracy and generalization of the trained model. Therefore, this section verifies the effectiveness and robustness of the proposed scheme in three different data distribution scenarios according to the experimental setup described in Section 6.1.

According to the experimental results shown in Table 5, it is obvious that when each party meets the local independent and identically distributed (IID) data, the proposed scheme has limited improvement on the accuracy of model training. Compared with the original algorithm, the proposed scheme only improves 1.05 and 1.17 percentage points respectively on the FashionMNIST and CIFAR-10 datasets. The reason is that in such an ideal federated learning environment, the randomly selected clients all have almost the same data distribution as the clients that satisfy the access policy.

Table 5. Training results under different independent identically distributed Settings.

Dataset		**FashionMNIST**			**CIFAR-10**		
		IID	$\alpha = 0.5$	$\alpha = 0.1$	**IID**	$\alpha = 0.5$	$\alpha = 0.1$
FedAvg	Average Accuracy	0.9118	0.8631	0.7522	0.8120	0.7046	0.6680
	Highest Accuracy	0.9127	0.8713	0.7811	0.8134	0.7121	0.6772
	ToA@0.85 ToA@0.7	24	175	-	66	669	-
ABEFedAvg	Average Accuracy	0.9223	0.8943	0.8303	0.8237	0.7508	0.7286
	Highest Accuracy	0.9228	0.8974	0.8389	0.8253	0.7583	0.7433
	ToA@0.85 ToA@0.7	22	62	-	52	167	237

However, it can be seen that when using the setting $\alpha = 0.5$ for Non-IID, the accuracy of the CNN model using ABEFedAvg is significantly higher than that of FedAvg with randomly selected clients, which is 3.12% and 4.62% higher for FashionMNIST and CIFAR-10 datasets, which shown in Figures 9 and 10, respectively. If we look at the more extreme case of $\alpha = 0.1$, the advantage of our scheme will be even more prominent, outdoing the random selection strategy in traditional federated learning by 7.81% and 6.06% in two datasets, respectively. The reason here is also obvious, because the proposed scheme can adaptively select participants with matching access policies in each round of training, which enables the system to control the data distribution of participants in a better range, so as to achieve higher training accuracy. It is worth mentioning that under the setting of $\alpha = 0.1$, due to the moderate access strategy used in this scheme, there may be a proportion that the number of selected parties is less than the default, but from the experimental results, the influence of this factor on the training accuracy is very limited. In addition, the '-' in Table 5 indicates that the algorithm cannot reach the target accuracy within a given number of rounds. For example, under the setting of $\alpha = 0.1$ of FashionMNIST dataset, neither algorithm can reach the test set accuracy of 0.85 within 500 communication rounds. Under the setting of $\alpha = 0.1$ of CIFAR-10 dataset, the traditional FedAvg algorithm cannot achieve an accuracy of 0.70 within 1000 communication rounds, while the ABEFedAvg algorithm can achieve the accuracy target with 237 communication rounds.

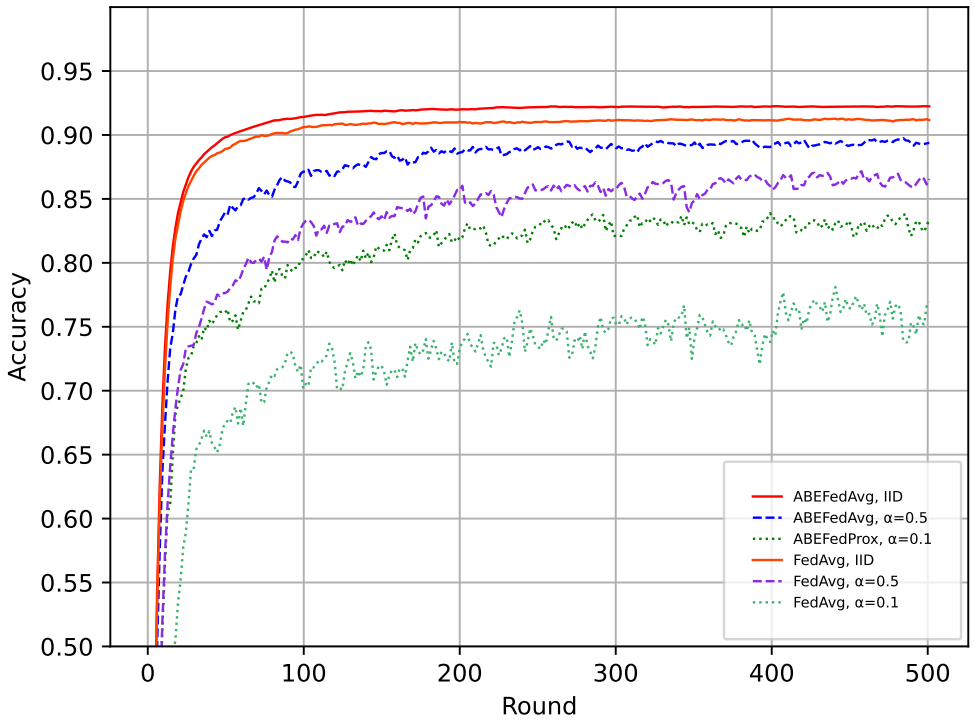

Figure 9. FashionMNIST for different IID degrees.

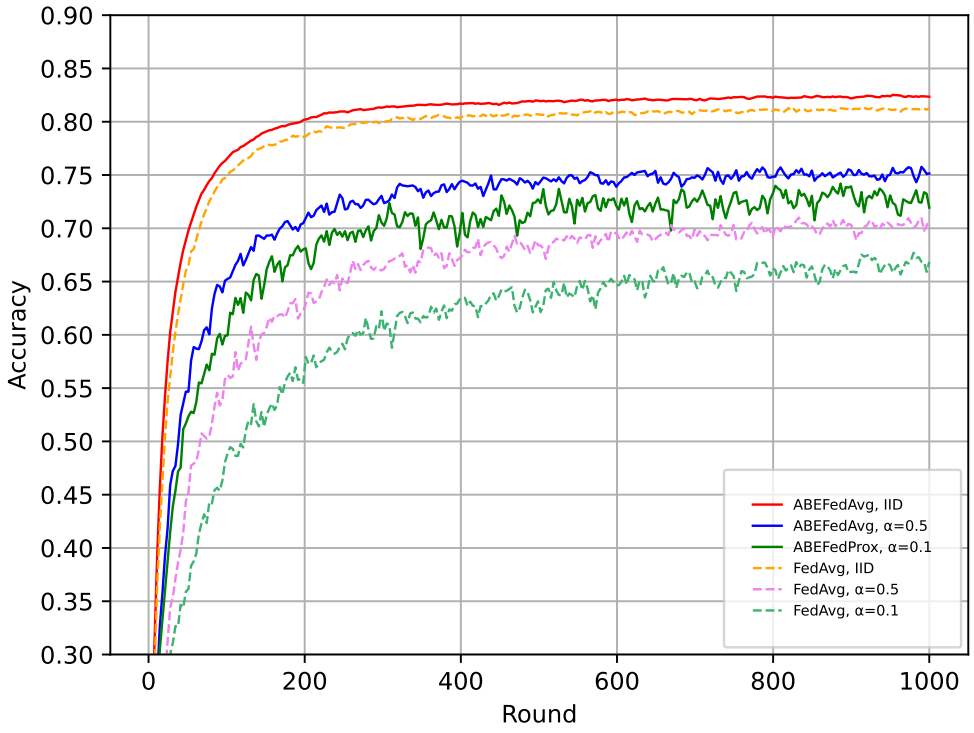

Figure 10. CIFAR-10 for different IID degrees.

6.2.3. Impact of Federated Learning Algorithms on Performance

This section investigates the applicability and optimization degree of the proposed attribute-based encryption party selection algorithm to two synchronous federated learning aggregation algorithms, FedProx and FedIR, when used as a module embeddable in federated learning. Although these latest schemes proposed many improvement strategies in the aggregation parameters, which improved the performance of the model to a certain

extent, most of them still used the random selection method to select participants, which had a great impact on the accuracy of the model. Therefore, this paper applies the client selection scheme as a couplable module to each mainstream algorithm to show its performance optimization effect for each aggregation strategy. Table 6 details the performance metrics for accuracy and processing time using two different datasets.

Table 6. Training results for different federated learning algorithms.

Algorithm	Average Accuracy		Highest Accuracy		ToA@0.85 ToA@0.7	
	F-MNIST	CIFAR-10	F-MNIST	CIFAR-10	F-MNIST	CIFAR-10
FedAvg	0.8631	0.7046	0.8713	0.7121	175	669
FedProx	0.8747	0.7100	0.8802	0.7192	143	401
FedIR	0.8786	0.7202	0.8827	0.7266	106	293
ABEFedAvg	0.8943	0.7508	0.8974	0.7583	70	167
ABEFedProx	0.8970	0.7597	0.9011	0.7666	61	146
ABEFedIR	0.9025	0.7725	0.9058	0.7803	51	125

Figures 11 and 12 show the training curves of each algorithm on FashionMNIST and CIFAR-10 datasets, respectively. It can be observed that the performance of different algorithms on the two datasets is basically the same. In general, the three algorithms can achieve the target accuracy within a given number of communication rounds, and FedAvg algorithm produces the lowest performance, followed by FedProx algorithm and FedIR algorithm. Although FedIR algorithm has higher accuracy, its training curve has a large degree of fluctuation due to the addition of additional weight information. For example, FedAvg using FashionMNIST dataset has an accuracy of 0.8631, while FedProx and FedIR have an accuracy of 0.8747 and 0.8786, respectively. After adding the attribute-based encryption selection module, it can be clearly seen that the performance of each algorithm is improved, and the accuracy is increased by 3.12, 2.23 and 2.39 percentage points respectively compared with the above three benchmark algorithms. Using the proposed scheme has the most obvious optimization effect on the FedAvg algorithm.

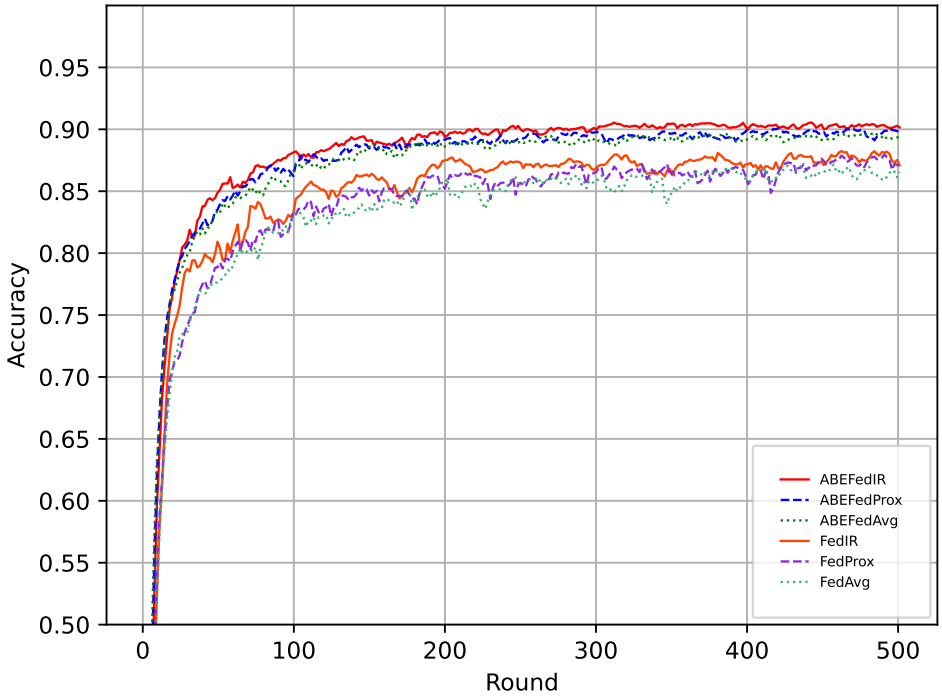

Figure 11. FashionMNIS Test accuracy for different federated learning algorithms.

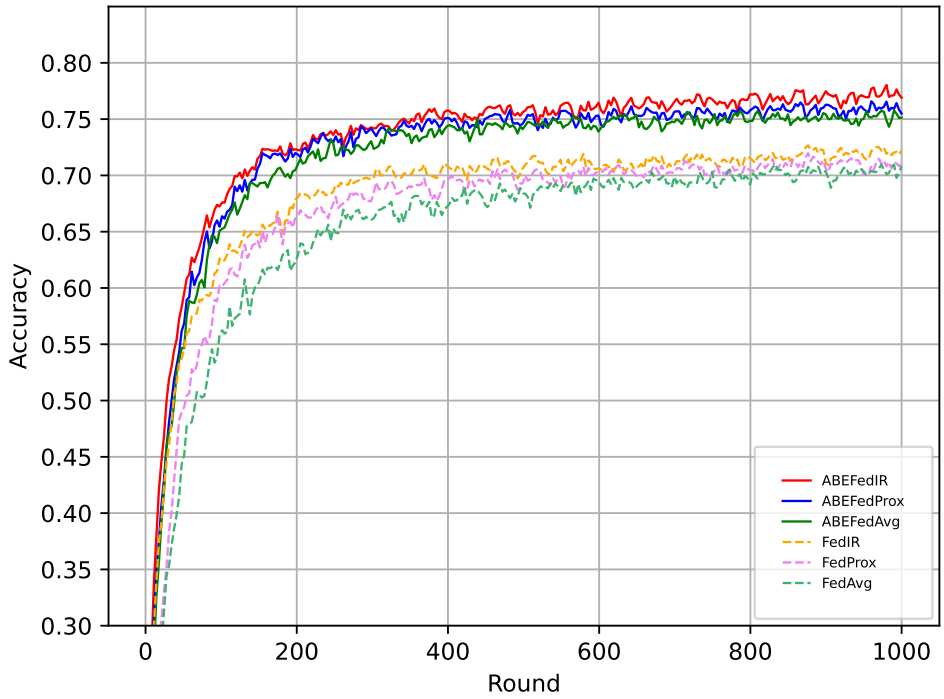

Figure 12. CIFAR-10 Test accuracy for different federated learning algorithms.

On the CIFAR-10 dataset, the proposed scheme can obtain more obvious advantages. The original FedAvg algorithm achieves an average accuracy of 0.7046 on this dataset, the FedProx algorithm is 0.7100, and the highest accuracy algorithm is FedIR, which reaches 0.7202. Using the proposed ABE can also improve the overall performance of the above algorithms on the test set. For example, for the CIFAR10 dataset, the accuracy of FedProx and FedIR algorithms with ABE filtering module is 0.7597 and 0.7725, respectively, which is 4.97 and 5.23 percentage points higher than that of the random selection scheme. In addition, although the introduction of encryption and decryption mechanism in the participant selection phase will increase the time overhead, the number of communication rounds can be greatly reduced once the appropriate participants are selected. The results show that the number of communication rounds is reduced by 502, 255 and 158 rounds respectively for the above three schemes. It can be concluded that the scheme in this paper has a strong optimization effect on various aggregation algorithms of synchronous federated learning.

6.2.4. Comparison with Other Participant Selection Schemes

The comparison between ABEFedAvg and other party selection schemes is shown in the related work section. The most successful recent works include Newt proposed by Zhao et al. [53] and FedFNS proposed by Wu et al. [45]. The former is to find the balance between accuracy and execution time in each round based on weight difference. The weight change between two adjacent rounds is defined as a utility that converges quickly. Moreover, since clients with large data volumes may negatively affect the training time, the ratio of the local dataset size to the total data size is also added as a coefficient of the client utility. Since it is not always necessary to select participants in each round of testing, the authors also designed a feedback control component that dynamically adjusts the frequency of customer selection; The latter is based on the selection of probability assignment, which designs an aggregation algorithm to determine the optimal subset of local model updates by excluding unfavorable local updates. In addition, a probabilistic node selection framework (FedPNS) was proposed, which dynamically adjusted the selection probability of the device according to its contribution to the data distribution model.

Next, the performance of the proposed scheme is compared with the above two latest federated learning participant selection schemes. Similarly, this section also uses the most classical FedAvg aggregation algorithm of federated learning to evaluate the test set accuracy and stability of the two datasets under the setting of $\mathcal{C} = 0.2$ and $\alpha = 0.5$. The experimental results are shown in Table 7. On the FashionMNIST dataset, the proposed attribute-based encryption access control scheme achieves an average accuracy of 0.8943, Zhao et al.'s scheme achieves an accuracy of 0.8782, and Wu et al.'s scheme achieves an accuracy of 0.8715. Compared with the above two schemes, the proposed scheme is improved by 1.83% and 2.62% respectively. On the CIFAR-10 dataset, the average accuracy of the proposed scheme reaches 0.7508, the other two schemes are 0.7294 and 0.7148, and the accuracy is improved by 2.93% and 5.04%, respectively. Then we further evaluate the number of communication rounds required by ABEFedAvg algorithm and other two schemes applied to federated learning training to achieve the target accuracy. As shown in Figures 13 and 14, on the FashionMNIST and CIFAR-10 datasets, the accuracy of 0.85 and 0.7 are achieved respectively, and the proposed scheme only needs 29 and 167 rounds. Although Newt and FedFNS have a great improvement over the original FedAvg random selection strategy, they are still weaker than the proposed FedABE scheme in this index. In summary, the party selection strategy based on attribute-based encryption proposed in this paper has obvious advantages even in the existing latest work, and has great application and promotion value.

Table 7. Training results for different participant selection schemes.

Algorithm	Average Accuracy		Highest Accuracy		ToA@0.85 ToA@0.7	
	Fashion MNIST	CIFAR-10	Fashion MNIST	CIFAR-10	Fashion MNIST	CIFAR-10
FedAvg	0.8631	0.7046	0.8713	0.7121	65	669
Newt [53]	0.8782	0.7294	0.8814	0.7353	39	213
FedFNS [45]	0.8715	0.7148	0.8766	0.7207	42	341
ABEFedAvg	0.8943	0.7508	0.8974	0.7583	29	167

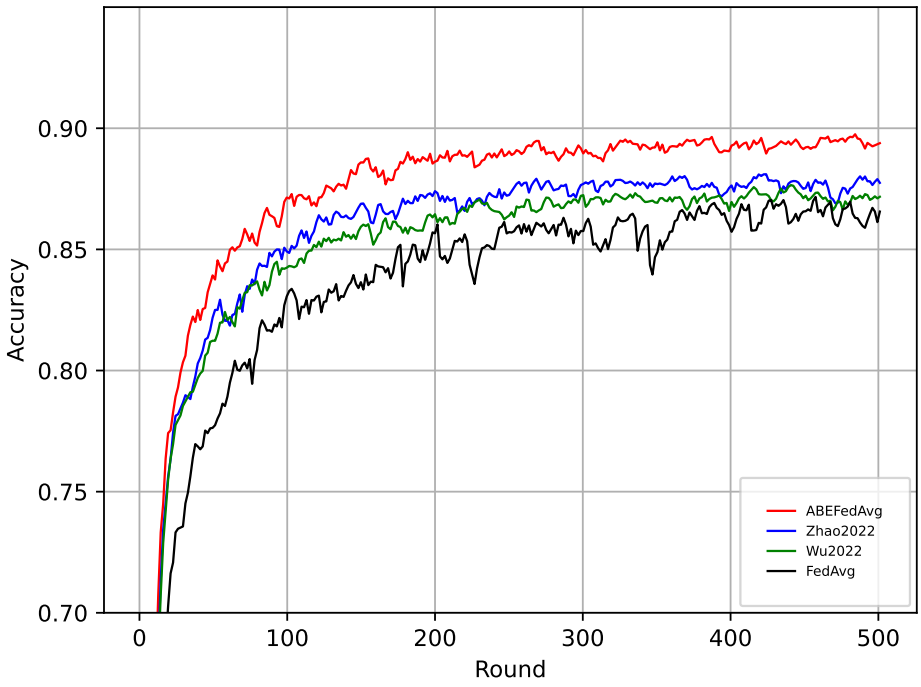

Figure 13. FashionMNIST Test accuracy for different participant selection schemes [45,53].

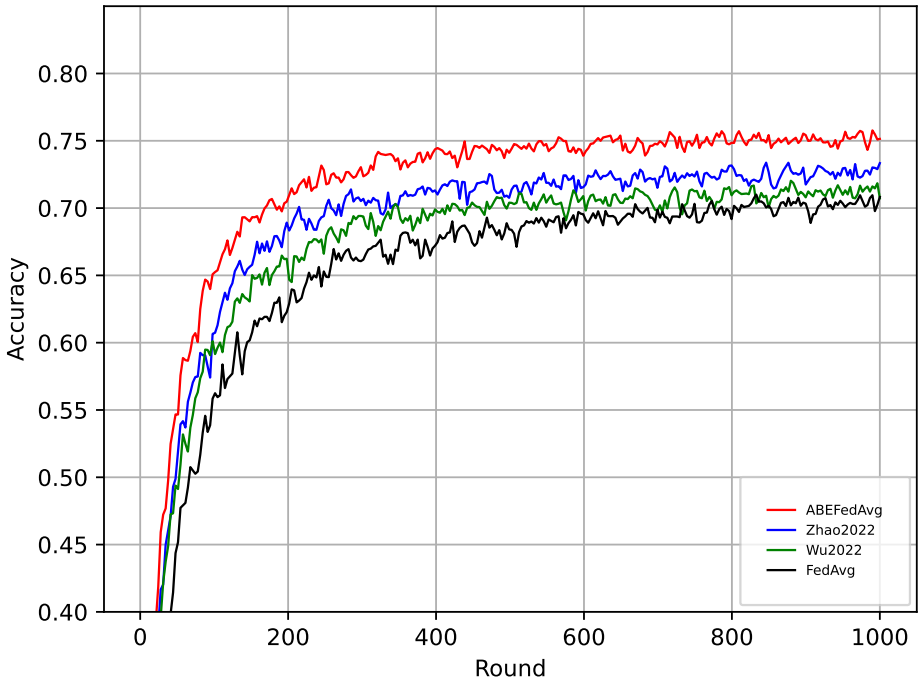

Figure 14. CIFAR-10 Test accuracy for different participant selection schemes [45,53].

7. Conclusions

In conclusion, our study introduces an innovative attribute-based participant selecting scheme for federated learning within smart city frameworks that leverages the integration of ciphertext-policy attribute-based encryption (CP-ABE) and consortium blockchain. This approach enhances both the security and efficiency of participant selection, mitigating common risks associated with privacy breaches and malicious attacks.

Our findings demonstrate that the proposed scheme significantly improves the efficiency of federated learning processes by enabling precise participant selection based on detailed attribute criteria, rather than relying on the traditional methods of random or resource-based selection. The attribute-based method ensures that only participants meeting specific pre-defined criteria contribute to the model training, thus optimizing the quality and relevance of the aggregated data.

Moreover, the incorporation of consortium blockchain technology provides a robust incentive mechanism and audit trail that ensures participant accountability and motivates continued engagement. This novel integration not only supports the scalability and sustainability of federated learning projects but also enhances their transparency and trustworthiness.

7.1. Theoretical and Practical Implications

Our research introduces a novel attribute-based participant selecting scheme enhanced with blockchain technology for federated learning in smart cities. This approach theoretically expands the understanding of federated learning by integrating privacy-preserving techniques (CP-ABE) and blockchain to safeguard against unauthorized access and ensure data integrity. Practically, the scheme provides a reliable and scalable solution for smart city administrators to deploy machine learning models that comply with stringent privacy regulations while maintaining high efficiency and participant motivation.

The implementation of our scheme in smart cities could significantly enhance the operational efficiency of various urban systems, such as public transportation networks, healthcare services, and emergency response systems. By ensuring that only qualified and authorized participants contribute to federated learning tasks, our model promotes the creation of more accurate and reliable predictive models, driving smarter decision-making in urban management.

7.2. Limitations

While our approach offers substantial improvements in privacy and efficiency, there are several limitations to consider. The complexity of CP-ABE may lead to an increased computational overhead, particularly as the number of attributes grows. This could potentially slow down the process in scenarios where real-time data processing is crucial. Additionally, our study's focus on theoretical design and simulated environments may not fully capture the practical challenges encountered in real-world implementations. The effectiveness and efficiency of the encryption might vary significantly under different operational conditions and with different data volumes.

7.3. Future Research Directions

Considering the identified limitations, future research should focus on optimizing the efficiency of attribute-based encryption techniques to reduce the computational demands, particularly in environments with extensive attributes. Further empirical research is also necessary to test the scheme across various real-world settings in smart cities, to evaluate its practicality and performance under diverse conditions. Such studies could help to refine the model, making it more robust and adaptable to different types of data and applications.

Exploring the application of our federated learning scheme in other domains, such as healthcare and public safety, could provide insights into its adaptability and effectiveness in other critical areas of smart city development. Moreover, integrating advanced machine learning techniques, such as deep learning, might enhance the predictive capabilities of the models trained using our scheme, thus broadening its applicability and impact.

Author Contributions: Conceptualization, X.Y. and H.Q.; methodology, X.Y. and H.Q.; software, X.Y. and H.Q.; validation, X.Y. and H.Q.; formal analysis, X.Y. and H.Q.; investigation, X.Y. and H.Q.; resources, X.Y. and H.Q.; data curation, X.Y. and H.Q.; writing—original draft preparation, X.Y. and H.Q.; writing—review and editing, X.W.; visualization, H.Q.; supervision, X.Z.; project administration, X.Z.; funding acquisition, X.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Key Research and Development Program of China under Grant 2020YFB2103803.

Data Availability Statement: The original contributions presented in the study are included in the article. Further inquiries can be directed to the corresponding authors.

Acknowledgments: We wish to acknowledge the anonymous referees who gave valuable suggestions to improve the work.

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

References

1. Hashem, I.A.T.; Usmani, R.S.A.; Almutairi, M.S.; Ibrahim, A.O.; Zakari, A.; Alotaibi, F.; Alhashmi, S.M.; Chiroma, H. Urban Computing for Sustainable Smart Cities: Recent Advances, Taxonomy, and Open Research Challenges. *Sustainability* **2023**, *15*, 3916. [CrossRef]
2. Band, S.S.; Ardabili, S.; Sookhak, M.; Theodore, A.; Elnaffar, S.; Moslehpour, M.; Csaba, M.; Torok, B.; Pai, H.T.; Mosavi, A. When Smart Cities Get Smarter via Machine Learning: An In-depth Literature Review. *IEEE Access* **2022**, *10*, 60985–61015. [CrossRef]
3. McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; Arcas, B.A.y. Communication-Efficient Learning of Deep Networks from Decentralized Data. In Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, Fort Lauderdale, FL, USA, 20–22 April 2017; pp. 1273–1282.
4. Liu, J.; Jia, J.; Che, T.; Huo, C.; Ren, J.; Zhou, Y.; Dai, H.; Dou, D. Fedasmu: Efficient asynchronous federated learning with dynamic staleness-aware model update. In Proceedings of the AAAI Conference on Artificial Intelligence, Vancouver, BC, Canada, 20–27 February 2024; pp. 13900–13908.
5. Abdelmoniem, A.M.; Sahu, A.N.; Canini, M.; Fahmy, S.A. Refl: Resource-efficient federated learning. In Proceedings of the Eighteenth European Conference on Computer Systems, Rome, Italy, 8–12 May 2023; pp. 215–232.
6. Xiong, Y.; Wang, R.; Cheng, M.; Yu, F.; Hsieh, C.J. Feddm: Iterative distribution matching for communication-efficient federated learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 16323–16332.

7. Chetoui, M.; Akhloufi, M.A. Peer-to-Peer Federated Learning for COVID-19 Detection Using Transformers. *Computers* **2023**, *12*, 106. [CrossRef]
8. Yang, H.; Ge, M.; Xue, D.; Xiang, K.; Li, H.; Lu, R. Gradient Leakage Attacks in Federated Learning: Research Frontiers, Taxonomy and Future Directions. *IEEE Netw.* **2023**, 1–8. [CrossRef]
9. Kairouz, P.; McMahan, H.B.; Avent, B.; Bellet, A.; Bennis, M.; Bhagoji, A.N.; Bonawitz, K.; Charles, Z.; Cormode, G.; Cummings, R.; et al. Advances and Open Problems in Federated Learning. *Found. Trends Mach. Learn.* **2021**, *14*, 1–210. [CrossRef]
10. Zhu, J.; Cao, J.; Saxena, D.; Jiang, S.; Ferradi, H. Blockchain-empowered federated learning: Challenges, solutions, and future directions. *ACM Comput. Surv.* **2023**, *55*, 1–31. [CrossRef]
11. Ali, A.; Ilahi, I.; Qayyum, A.; Mohammed, I.; Al-Fuqaha, A.; Qadir, J. A systematic review of federated learning incentive mechanisms and associated security challenges. *Comput. Sci. Rev.* **2023**, *50*, 100593. [CrossRef]
12. Chai, Z.; Ali, A.; Zawad, S.; Truex, S.; Anwar, A.; Baracaldo, N.; Zhou, Y.; Ludwig, H.; Yan, F.; Cheng, Y. TiFL: A Tier-based Federated Learning System. In Proceedings of the HPDC '20: The 29th International Symposium on High-Performance Parallel and Distributed Computing, Stockholm, Sweden, 23–26 June 2020; pp. 125–136. [CrossRef]
13. Marnissi, O.; Hammouti, H.E.; Bergou, E.H. Client selection in federated learning based on gradients importance, NY, USA. In Proceedings of the Ninth International Conference on Modeling, Simulation and Applied Optimization, Marrakesh, Morocco, 26–28 April 2023.
14. Zhang, W.; Wang, X.; Zhou, P.; Wu, W.; Zhang, X. Client Selection for Federated Learning with Non-IID Data in Mobile Edge Computing. *IEEE Access* **2021**, *9*, 24462–24474. [CrossRef]
15. Ozdayi, M.S.; Kantarcioglu, M.; Gel, Y.R. Defending against backdoors in federated learning with robust learning rate. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 9268–9276.
16. Nagalapatti, L.; Narayanam, R. Game of gradients: Mitigating irrelevant clients in federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Virtually, 2–9 February 2021; Volume 35, pp. 9046–9054.
17. Zhang, B.; Lu, G.; Qiu, P.; Gui, X.; Shi, Y. Advancing Federated Learning through Verifiable Computations and Homomorphic Encryption. *Entropy* **2023**, *25*, 1550. [CrossRef]
18. Shen, X.; Jiang, H.; Chen, Y.; Wang, B.; Gao, L. Pldp-fl: Federated learning with personalized local differential privacy. *Entropy* **2023**, *25*, 485. [CrossRef]
19. Wu, X.; Huang, F.; Hu, Z.; Huang, H. Faster adaptive federated learning. In Proceedings of the AAAI Conference on Artificial Intelligence, Washington, DC, USA, 7–14 February 2023; pp. 10379–10387.
20. Feng, D.; Helena, C.; Lim, W.Y.B.; Ng, J.S.; Jiang, H.; Xiong, Z.; Kang, J.; Yu, H.; Niyato, D.; Miao, C. CrowdFL: A Marketplace for Crowdsourced Federated Learning. In Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, Virtual Event, 22 February–1 March 2022; pp. 13164–13166.
21. Zhang, Y.; Deng, R.H.; Xu, S.; Sun, J.; Li, Q.; Zheng, D. Attribute-based encryption for cloud computing access control: A survey. *ACM Comput. Surv. (CSUR)* **2020**, *53*, 1–41. [CrossRef]
22. Lai, F.; Zhu, X.; Madhyastha, H.V.; Chowdhury, M. Oort: Informed Participant Selection for Scalable Federated Learning. *arXiv* **2020**, arXiv:2010.06081.
23. Li, C.; Zeng, X.; Zhang, M.; Cao, Z. PyramidFL: A fine-grained client selection framework for efficient federated learning. In Proceedings of the 28th Annual International Conference on Mobile Computing and Networking, Sydney, Australia, 17–21 October 2022; pp. 158–171.
24. Wang, H.; Kaplan, Z.; Niu, D.; Li, B. Optimizing federated learning on non-iid data with reinforcement learning, Toronto, ON, Canada. In Proceedings of the IEEE INFOCOM 2020, Toronto, ON, Canada, 6–9 July 2020; pp. 1698–1707.
25. Sarikaya, Y.; Ercetin, O. Motivating workers in federated learning: A stackelberg game perspective. *IEEE Netw. Lett.* **2019**, *2*, 23–27. [CrossRef]
26. Richardson, A.; Filos-Ratsikas, A.; Faltings, B. Rewarding high-quality data via influence functions. *arXiv* **2019**, arXiv:1908.11598.
27. Xu, J.; Wang, C.; Jia, X. A survey of blockchain consensus protocols. *ACM Comput. Surv.* **2023**, *55*, 1–35. [CrossRef]
28. Almutairi, W.; Moulahi, T. Joining Federated Learning to Blockchain for Digital Forensics in IoT. *Computers* **2023**, *12*, 157. [CrossRef]
29. Weng, J.; Weng, J.; Zhang, J.; Li, M.; Zhang, Y.; Luo, W. Deepchain: Auditable and privacy-preserving deep learning with blockchain-based incentive. *IEEE Trans. Dependable Secur. Comput.* **2019**, *18*, 2438–2455. [CrossRef]
30. Bao, X.; Su, C.; Xiong, Y.; Huang, W.; Hu, Y. Flchain: A blockchain for auditable federated learning with trust and incentive. In Proceedings of the 2019 5th International Conference on Big Data Computing and Communications (BIGCOM), Qingdao, China, 9–11 August 2019; pp. 151–159.
31. Sahai, A.; Waters, B.R. Fuzzy Identity-Based Encryption. In Proceedings of the 24th annual international conference on Theory and Applications of Cryptographic Techniques, Zurich, Switzerland, 26–30 May 2004.
32. Bethencourt, J.; Sahai, A.; Waters, B. Ciphertext-Policy Attribute-Based Encryption. In Proceedings of the IEEE Symposium on Security & Privacy, Berkeley, CA, USA, 20–23 May 2007.
33. Emura, K.; Miyaji, A.; Nomura, A.; Omote, K.; Soshi, M. A ciphertext-policy attribute-based encryption scheme with constant ciphertext length. In Proceedings of the International Conference on Information Security Practice and Experience, Xi'an, China, 13–15 April 2009; pp. 13–23.

34. Waters, B. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Proceedings of the International Workshop on Public Key Cryptography, Taormina, Italy, 6–9 March 2011; pp. 53–70.
35. Pirretti, M.; Traynor, P.; McDaniel, P.; Waters, B. Secure attribute-based systems. *J. Comput. Secur.* **2010**, *18*, 799–837. [CrossRef]
36. Zhang, Y.; Chen, X.; Li, J.; Li, H.; Li, F. FDR-ABE: Attribute-based encryption with flexible and direct revocation. In Proceedings of the 2013 5th International Conference on Intelligent Networking and Collaborative Systems, Xi'an, China, 9–11 September 2013; pp. 38–45.
37. Hur, J.; Noh, D.K. Attribute-based access control with efficient revocation in data outsourcing systems. *IEEE Trans. Parallel Distrib. Syst.* **2010**, *22*, 1214–1221. [CrossRef]
38. Li, J.; Yao, W.; Han, J.; Zhang, Y.; Shen, J. User collusion avoidance CP-ABE with efficient attribute revocation for cloud storage. *IEEE Syst. J.* **2017**, *12*, 1767–1777. [CrossRef]
39. Prantl, T.; Zeck, T.; Horn, L.; Iffländer, L.; Bauer, A.; Dmitrienko, A.; Krupitzer, C.; Kounev, S. Towards a Cryptography Encyclopedia: A Survey on Attribute-Based Encryption. *J. Surveill. Secur. Saf.* **2023**, *4*, 129–154. [CrossRef]
40. Tseng, Y.F.; Huang, J.J. Cryptanalysis on Two Pairing-Free Ciphertext-Policy Attribute-Based Encryption Schemes. In Proceedings of the 2020 International Computer Symposium (ICS), Tainan, Taiwan, 17–19 December 2020; pp. 403–407.
41. Ding, S.; Li, C.; Li, H. A novel efficient pairing-free CP-ABE based on elliptic curve cryptography for IoT. *IEEE Access* **2018**, *6*, 27336–27345. [CrossRef]
42. Wang, Y.; Chen, B.; Li, L.; Ma, Q.; Li, H.; He, D. Efficient and secure ciphertext-policy attribute-based encryption without pairing for cloud-assisted smart grid. *IEEE Access* **2020**, *8*, 40704–40713. [CrossRef]
43. Nishio, T.; Yonetani, R. Client selection for federated learning with heterogeneous resources in mobile edge. In Proceedings of the ICC 2019-2019 IEEE international conference on communications (ICC), Shanghai, China, 20–24 May 2019; pp. 1–7.
44. Cho, Y.J.; Wang, J.; Joshi, G. Client selection in federated learning: Convergence analysis and power-of-choice selection strategies. *arXiv* **2020**, arXiv:2010.01243.
45. Wu, H.; Wang, P. Node selection toward faster convergence for federated learning on non-iid data. *IEEE Trans. Netw. Sci. Eng.* **2022**, *9*, 3099–3111. [CrossRef]
46. Song, T.; Tong, Y.; Wei, S. Profit allocation for federated learning. In Proceedings of the 2019 IEEE International Conference on Big Data (Big Data), Los Angeles, CA, USA, 9–12 December 2019; pp. 2577–2586.
47. Yu, H.; Liu, Z.; Liu, Y.; Chen, T.; Cong, M.; Weng, X.; Niyato, D.; Yang, Q. A sustainable incentive scheme for federated learning. *IEEE Intell. Syst.* **2020**, *35*, 58–69. [CrossRef]
48. Zeng, R.; Zhang, S.; Wang, J.; Chu, X. Fmore: An incentive scheme of multi-dimensional auction for federated learning in mec. In Proceedings of the 2020 IEEE 40th International Conference on Distributed Computing Systems (ICDCS), Singapore, 29 November–1 December 2020; pp. 278–288.
49. Zhan, Y.; Li, P.; Qu, Z.; Zeng, D.; Guo, S. A learning-based incentive mechanism for federated learning. *IEEE Internet Things J.* **2020**, *7*, 6360–6368. [CrossRef]
50. Li, T.; Sahu, A.K.; Zaheer, M.; Sanjabi, M.; Talwalkar, A.; Smith, V. Federated optimization in heterogeneous networks. *Proc. Mach. Learn. Syst.* **2020**, *2*, 429–450.
51. Hsu, T.M.H.; Qi, H.; Brown, M. Federated visual classification with real-world data distribution. In Proceedings of the Computer Vision—ECCV 2020: 16th European Conference, Glasgow, UK, 23–28 August 2020; pp. 76–92.
52. Hsu, T.M.H.; Qi, H.; Brown, M. Measuring the effects of non-identical data distribution for federated visual classification. *arXiv* **2019**, arXiv:1909.06335.
53. Zhao, J.; Chang, X.; Feng, Y.; Liu, C.H.; Liu, N. Participant selection for federated learning with heterogeneous data in intelligent transport system. *IEEE Trans. Intell. Transp. Syst.* **2022**, *24*, 1106–1115. [CrossRef]