


Article

# Trustworthy Anti-Collusion Federated Learning Scheme Optimized by Game Theory

Qiuxian Li <sup>1,2,†</sup> , Quanxing Zhou <sup>1,2,\*,†</sup>, Mingyang Li <sup>2</sup> and Zhenlong Wang <sup>3</sup><sup>1</sup> College of Big Data Engineering, Kaili University, Kaili 556011, China; qiuxianll@163.com<sup>2</sup> College of Information, St. Paul University Philippines, Tuguegarao City 3500, Cagayan, Philippines; myli1992@126.com<sup>3</sup> College of Microelectronics and Artificial Intelligence, Kaili University, Kaili 556011, China; xskcxcy@163.com

\* Correspondence: aqxgzs@163.com

† These authors contributed equally to this work.

**Abstract:** Federated learning, a decentralized paradigm, offers the potential to train models across multiple devices while preserving data privacy. However, challenges such as malicious actors and model parameter leakage have raised concerns. To tackle these issues, we introduce a game-theoretic, trustworthy anti-collusion federated learning scheme, which combines game-theoretic techniques and rational trust models with functional encryption and smart contracts for enhanced security. Our empirical evaluations, using datasets like MNIST, CIFAR-10, and Fashion MNIST, underscore the influence of data distribution on performance, with IID setups outshining non-IID ones. The proposed scheme also showcased scalability across diverse client counts, adaptability to various tasks, and heightened security through game theory. A critical observation was the trade-off between privacy measures and optimal model performance. Overall, our findings highlight the scheme's capability to bolster federated learning's robustness and security.

**Keywords:** federated learning; game theory; rational trust model; smart contract; functional encryption



**Citation:** Li, Q.; Zhou, Q.; Li, M.; Wang, Z. Trustworthy Anti-Collusion Federated Learning Scheme Optimized by Game Theory. *Electronics* **2023**, *12*, 3867. <https://doi.org/10.3390/electronics12183867>

Academic Editor: Aryya Gangopadhyay

Received: 18 August 2023

Revised: 9 September 2023

Accepted: 11 September 2023

Published: 13 September 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

With the rapid development of the era of big data and the internet, the issue of privacy leakage of massive user information and data [1,2] has become increasingly prominent. The exponential growth in data, combined with their diverse nature, has posed significant challenges in terms of data privacy and security. As a novel distributed machine learning framework, federated learning [3] allows data to be stored on local mobile devices, coordinating model parameters across these devices to aggregate the model. This unique approach ensures that data remain decentralized, eliminating the need for centralized servers that often pose security risks. During federated learning, each mobile device uses local data for private training, eliminating the need for personal privacy data to be transmitted over the network. This method, to some extent, protects the data privacy of each mobile device and effectively addresses the “data island” problem [4]. Consequently, more and more researchers are beginning to focus on the security and privacy of federated learning [5,6].

While federated learning offers a solution to some of the data privacy issues, ensuring the active participation of devices in the learning process remains a challenge. Devices incur training costs and communication overheads, which can sometimes discourage them from actively participating. In order to encourage the active participation of mobile devices in data sharing and improve the quality and efficiency of federated learning model training, numerous research solutions have been proposed and widely applied. For instance, Konečný et al. [7] proposed two methods to reduce the cost of uplink communication, reducing the communication cost of federated learning by two orders of magnitude. Kim et al. [8] introduced the architecture of Blockchain Federated Learning (BlockFL), in which local

learning model updates can be exchanged and verified. Hardy et al. [9] realized privacy data protection for each peer mobile device and provided a significantly enhanced federated learning environment for all mobile devices. Li et al. [10] provided an asymptotically tight lower bound for the goals that communication compression may achieve. These research solutions, though innovative, often operate under the assumption that participating nodes are honest, which may not always be the case in real-world scenarios. Considering the heterogeneity of channel performance and the competitive relationship of data transmission on wireless channels in the federated learning framework, Zhao et al. [11] proposed a new group asynchronous model synchronization method, significantly improving the training efficiency of federated learning. Chen et al. [12] reduced the communication cost of each parameter of federated learning model training to below 1.78 bits through linear techniques based on sparse random projection, thus improving the learning efficiency of federated learning.

To address the challenges posed by potential dishonest nodes and to ensure optimal participation, game theory offers a promising approach. Katz et al. [13] explored the relationship between game theory and security protocols, especially the problem of security protocols among distrustful participants. Game theory, with its ability to model and predict the behavior of rational agents, can be effectively utilized in the context of federated learning. Many scholars have introduced the idea of game theory into federated learning schemes, using participants' rationality as a starting point to design reasonable utility functions to motivate participants to actively participate in model training. For example, He et al. [14] proposed a new federated learning incentive model that encourages mobile nodes to participate in training tasks by maximizing collective utility functions. Martinez et al. [15] proposed a distributed learning framework based on blockchain design to ensure data security, and made payments for gradient upload based on a new index of validation errors, effectively enhancing the enthusiasm of each node for model training. Zhou et al. [16] used game theory and Micali–Rabin random vector representation technology to improve the communication efficiency of model training and ensured that all rational participants could obtain optimal utility returns. Zhu et al. [17] built a decentralized parameter aggregation chain from the centralized parameter server in federated learning, incentivizing collaborating nodes to verify model parameters, enhancing trust between nodes, and thus improving the efficiency of federated learning. Stergiou et al. [18] proposed a novel architectural scenario based on cloud computing, leveraging the innovative models of federated learning. Their proposed model aims to provide users with a more energy-efficient system architecture and environment, with the objective centered around data management. Wassan et al. [19] introduced differential privacy in federated learning, employing the adaptive GBTM model algorithm for local updates. This approach aids in adjusting model parameters based on data characteristics and gradients.

Nevertheless, while the integration of game theory provides a mechanism to incentivize honest participation, there is still a need to address the communication overhead of federated learning. In order to effectively balance the relationship between model parameter privacy security, utility returns of all rational participants, and communication overhead during the federated learning process, we construct a new federated learning scheme using game theory, smart contracts, rational trust models, and function encryption techniques.

The specific work includes:

- Introducing local training rational participants using game theory and rational trust models, reducing communication frequency, and constructing a trustworthy anti-collusion federated learning game scheme to reduce communication costs;
- Using blockchain networks and smart contract technology to ensure no new malicious nodes can participate in model training during the federated learning process, and recording all participants' trust values and transaction processes through smart contracts;
- Ensuring model parameter privacy security through function encryption technology and achieving privacy-protected sharing between task publishers and data owners;

- Proving the correctness and security of the scheme through analysis and experimental simulation, with empirical evidence showing that the scheme does, indeed, improve the learning efficiency of federated learning.

This paper is structured as follows. Following the introduction, we delve into the preliminary knowledge in Section 2. In Section 3, we introduce the “Credible Defense Against Collusion Game Model”. Section 4 presents a detailed discussion of the “Credible Anti-Collusion Federal Learning Scheme”. In Section 5, we analyze the proposed scheme. The experimental results are presented in Section 6. Finally, we conclude the paper and discuss future prospects in Section 7.

## 2. Preliminary Knowledge

### 2.1. Game Theory

**Definition 1** (Game Theory). *The basic structure of game theory is composed of three main elements: the set of players  $P$ , the strategy space  $S$ , and the utility function  $U$ . Specifically, where  $G = \{P, S, u\}$ ,  $P = \{P_1, \dots, P_n\}$ ,  $S = \{S_1, \dots, S_n\}$ ,  $u = \{u_1, \dots, u_n\}$ . The utility function:  $u_i : S \rightarrow R$  (where  $R$  represents the real number space), is used to describe the benefit level of the player  $i$  under various strategy combinations. An extensive form game can be seen as a four-tuple  $(P, W, (I_i)_i \in P, (\leq_i)_i \in P)$ , specifically:*

- $P$  represents the set of all rational participants in the agreement;
- $W$  is the set of action sequences of the participants, satisfying the following properties:
  1. Includes empty sequence  $\emptyset \in W$ ;
  2. If sequence  $(a_k)_{k=1}^g \in W$  exists and  $0 < v < g$ , then sequence  $(a_k)_{k=1}^v \in W$  also exists;
  3. If for any positive integer  $v$ , the infinite action sequence  $(a_k)_{k=1}^\infty$  satisfies condition  $(a_k)_{k=1}^v \in W$ , then sequence  $(a_k)_{k=1}^\infty \in W$  exists.
- $I_i$  represents the information set of the rational participant  $i \in P$ , i.e., the information that the rational participant knows or understands before making a strategy choice action;
- $\leq_i$  represents the preference relations of the rational participants, i.e., in the protocol scheme, each rational participant  $i \in P$  has a preference relation on the non-terminal sequences.

**Definition 2** (Nash Equilibrium). *For game  $G = \{P, S, u\}$ , if there exists a strategy  $s^* = (s_1^*, \dots, s_n^*)$  combination composed of some strategy of all game parties, where in any game party  $P_i$  strategy  $s_i^*$ , is the best strategy when dealing with other game parties' strategy combination  $(s_1^*, \dots, s_n^*)$ . In other words, for all  $s_j \in S$ , there exists a game  $u_i(s_i^*) \geq u_i(s_j^*, s_{-i}^*)$ , for any strategy  $s_{ij} \in S$ , all meet the condition that strategy  $(s_1^*, \dots, s_n^*)$  is the optimal choice of game  $G$ , then we call this strategy combination  $(s_1^*, \dots, s_n^*)$  a Nash equilibrium of game  $G$ .*

In our approach, we adopt the Nash Equilibrium as it symbolizes a fixed and consistent strategy combination. Within this equilibrium, every participant's strategy is the optimal choice in relation to the strategies of other participants. In such a balanced state, no participant has the motivation to unilaterally alter their strategy. This provides a solid benchmark for multiple participants in federated learning, ensuring the overall system's stability and efficient operation. While the Nash Equilibrium might not always achieve a global optimum, it serves as a practical and operational strategy in a scenario characterized by multi-party interactions and decentralized decision-making.

### 2.2. Function Encryption

Function encryption [20] is a complex encryption technique that extends traditional public key encryption algorithms, allowing authorized parties to selectively compute on ciphertext and directly extract the results. A comprehensive function encryption algorithm typically consists of the following five parts:

1. System initialization algorithm *Setup*, used to generate system model parameters and the prime group  $G$  needed by the system;

2. Key generation algorithm *MasterKeyGeneration*, responsible for generating the system’s master key, which includes the master public key *mpk* and master private key *msk*;
3. Function generation algorithm *FunctionKeyDerivation*, by using the system function *F* and master private *msk* key as input to generate the private key *sk<sub>m</sub>* of the encryption function, thereby obtaining function results *F*(·);
4. Encryption algorithm *Encryption*, used to encrypt sensitive data *m*, and thus producing ciphertext *C*(*m*);
5. Decryption algorithm *Decryption*, by using ciphertext *C*(*m*) and the corresponding private key *sk<sub>m</sub>* as input, the function decryption result *F*(*m*) can be obtained.

2.3. Rational Trust Model

The rational trust model [21] is a quantifiable trust model that combines the ideas of game theory and trust management to design a trust measurement function or model in a protocol with multiple rational parties *n*. In the rational trust model, by designing an appropriate utility function, all rational participants are encouraged to actively execute the protocol. The following is a definition of a basic trust function:

**Definition 3** (Trust Function). Suppose  $\zeta_i^N$  represents the trust value of the participant  $P_i$  in cycle  $N$ , where  $-1 \leq \zeta_i^N \leq +1$ , and  $\zeta_i^N = 0$  is the trust value of the new parameter participant  $P_i$ . A trust function is a mapping from  $R \times N$  to  $R$ , that is,  $(\zeta_i^{N-1}, \alpha_i) \mapsto \zeta_i^N$ , where  $\zeta_i^{N-1}$  represents the trust value of participant  $P_i$  in cycle  $N - 1$ ; and  $\alpha_i \in (0, 1)$  indicates whether participant  $P_i$  abides by the protocol in cycle  $N$ , that is,  $\alpha_i = 1$  represents the participant that honestly executes the protocol, and  $\alpha_i = 0$  represents the participant betraying the protocol. Therefore, the trust function can be formalized as:

$$f_1 : ((f_1 : (\zeta_i^{N-1}, \alpha_i) \mapsto \zeta_i^N). \tag{1}$$

**Definition 4** (Rational Trust Function). Since all participants in the scheme are rational and typically choose the optimal strategy to maximize their own benefits, it is necessary to introduce appropriate parameters in the trust model to motivate rational participants, to resist various malicious attacks, and to improve the trust value of participants. Through the analysis of rational participants, we introduce parameter  $l_i$  into the trust function to reconstruct the rational trust function:  $a = a$  (2)

$$f_2 : (\zeta_i^{N-1}, \alpha_i, l_i) \mapsto \zeta_i^N. \tag{2}$$

where  $l_i \geq 0$  represents the lifespan of each rational mobile device end in the scheme. When participants choose to betray the protocol, they will be punished and the protocol will be terminated. This rational participant will lose all original trust values, and their lifespan and trust value will start again from zero. It is difficult to regain the trust of other rational participants, thus losing the opportunity to participate in tasks again.

3. Credible Defense against Collusion Game Model

We have constructed a credible game model for federated learning that defends against collusion, ingeniously integrating the historical trust values of rational participants with the utility derived from completing training tasks. The goal is to delve deep into the learning efficiency of federated learning and to elevate the quality and accuracy of the training of federated learning models. This credible defense against collusion game model extensively draws from traditional federated learning and game theory. Starting from the perspective of participants’ self-interest, a logical and effective utility function is designed to motivate rational participants to actively engage in model training. In this model, every rational participant adopts respective behavioral strategies aiming to maximize their benefits. Any behavior that deviates from the established agreement will be met with stringent penalties and, concurrently, the individual’s historical trust value will be entirely reset.

In this context, smart contract technology plays a pivotal role. Initially, both task publishers and data owners establish the parameters and execution standards for federated learning tasks through smart contracts. This ensures the transparency and fairness of the tasks. When data owners complete model training and upload their results, the smart contract automatically verifies these results and, based on pre-defined criteria, metes out rewards or penalties. This not only obviates the need for manual verification but also guarantees that every participant acts in accordance with the agreement. Additionally, the smart contract logs the historical trust value of each data owner, providing task publishers with a continuously updated reference for trustworthiness. This process further strengthens the fairness and transparency of the system.

A schematic representation of the model participants is shown in Figure 1. The specific steps of the model are as follows:

- Step 1:** Evaluate the historical trust value of data owners. During the model training phase, the task publisher lacks direct insight into each data owner’s work ethic. Thus, to predict the accuracy of model parameters post-task completion, the task publisher participating in federated learning needs to evaluate the historical trust value of each data owner. By opting for data owners with high historical trust values for model training, the learning efficiency of federated learning can be further enhanced.
- Step 2:** Execute federated learning tasks. The task publisher posts the global model training parameters to the blockchain and then selects apt data owners for executing model training tasks based on the analysis of historical trust values. Upon receiving the model parameters, data owners conduct private training based on their individual data and upload the training outcomes to the blockchain. Subsequently, the task publisher verifies the task results and evaluates the accuracy of the data owner’s work during the task’s execution.
- Step 3:** Analyze the execution task results. Data owners upload the updated model training parameters to the blockchain, after which the task publisher verifies these updated parameters. Given that function encryption technology is employed to safeguard the security of model parameters during task publication and verification, it becomes imperative to validate the authenticity of the updated model parameters.
- Step 4:** Compute the utility of rational participants. Once the task issuer retrieves the optimally updated global model parameters, the federated learning training task concludes. At this juncture, the utility function of rational participants must be computed. Analyzing the utility of each participant allows for gauging the efficiency and accuracy of the task.

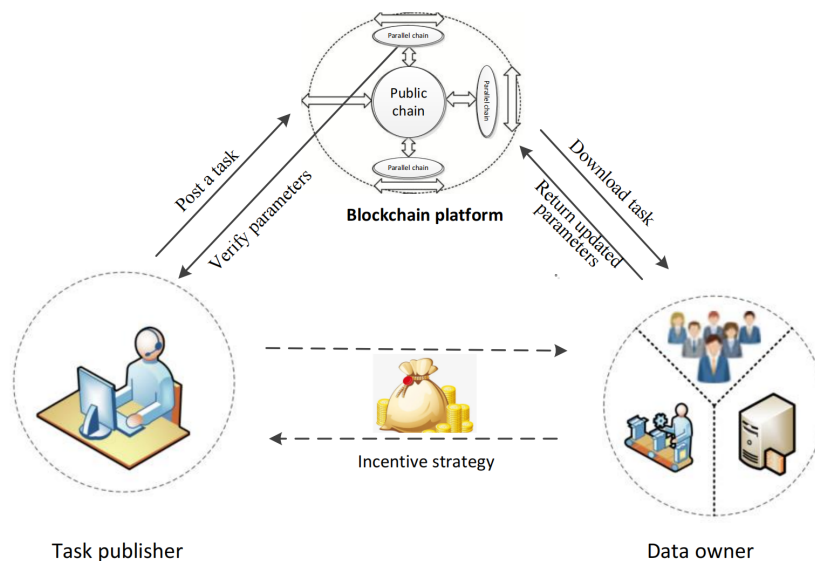


Figure 1. Schematic diagram of model participants.



### 3.1. Model Parameters

The main parameters involved in the game-theoretic trustworthy anti-collusion federated learning model proposed in this paper are shown in Table 1 and described in detail in the text.

**Table 1.** The parameters used in this model and their meanings.

Parameter	Meaning
$d_a$	Deposit by the task publisher
$d_{p_i}$	Deposit by the data owner
$d_o$	Deposit by the colluding owner
$a$	Reward for optimal global parameters by the publisher
$c_i$	Data owner's model training cost
$w$	Reward to honest trainer by the publisher
$r_i$	Payment to data owner for training
$f_i$	Penalty for protocol breaches
$v$	Model parameter verification cost
$t_1$	Task publishing deadline
$t_2$	Collusion initiation deadline
$t_3$	Model training task deadline

In the model, as all participants are rational, they will choose the optimal behavioral strategy to maximize their own benefits. Therefore, the credible defense against the collusion game model needs to maintain balance, and the parameters in the model should satisfy the following relationships:

- There exists  $a - nr_i > 0$ , otherwise the task publisher has no motivation to publish federated learning tasks;
- There exists  $r_i > c_i$ , otherwise the data owner has no motivation to accept model training tasks;
- There exist  $d_A > a$  and  $d_{p_i} > r_i$ ;  $d_o > r_i$  and  $f_i > d_A$ , because only when the deposit and fine are large enough can participants be encouraged to follow the protocol execution honestly;
- There exists  $d_o < d_{p_i}$ , otherwise the data owner has the motivation to initiate a collusion strategy.

In the credible defense against the collusion game model of federated learning, since every rational participant is selfish and strives to maximize their own benefits, the personal utility of the participants is determined by their own behavioral strategies and the behavioral strategies of other participants in the model. Therefore, an effective credible defense against the collusion game model needs to reasonably design the rational participants, feasible strategies, utility functions, and anti-collusion mechanisms in the model to encourage all rational participants to actively participate in model training.

### 3.2. Participants

The primary entities to be modeled in the credible defense against the collusion game model are the rational participants within the model. This game model mainly involves two types of participants: one is the task publishers  $A$  who publish federated learning tasks, and the other is the mobile devices or data owners  $P_i$  who perform model training tasks. As all participants are rational, task publishers  $A$  seek to maximize their own interests while ensuring optimal global parameters for model training. Similarly, data owners  $P_i$  also seek to maximize their own interests while meeting the task execution requirements of the task publishers. Therefore, the set of participants in this model can be formally defined as  $P = \{A, P_i\}$ .

### 3.3. Feasible Strategies

In this model, we assume that the behavioral strategy set of the rational task publishers  $A$  is  $s_a = \{s_{a1}, s_{a2}\}$ , where  $s_{a1}$  denotes the behavioral strategy of “incentivizing” data owners and is assigned a value of 1;  $s_{a2}$  represents the strategy of “not incentivizing” data owners and is assigned a value of 0. Similarly, the behavior strategy set for selfish rational data owners  $P_i$  is  $s_{p_i} = \{s_{p_{i1}}, s_{p_{i2}}\}$ , where  $s_{p_{i1}}$  represents the “honest” strategy with a value of 1, and  $s_{p_{i2}}$  represents the “collusion” strategy with a value of 0.

In the credible defense against collusion game model, we assume that task publishers  $A$  first adopt a behavioral strategy to decide whether to incentivize data owners  $P_i$  to honestly perform model parameter training tasks. Subsequently, data owners  $P_i$  will adopt corresponding strategies to maximize their personal interests based on the behavioral strategy of task publishers  $A$ . Therefore, this game model is an asymmetric information game, in which each rational participant chooses the appropriate behavior strategy to update their local state and optimize their personal utility based on different sets of information.

### 3.4. Utility Functions

In this model, as all participants are rational, task publishers  $A$  always hope that data owners  $P_i$  will honestly use their own data to perform model parameter training tasks; data owners  $P_i$  always hope that task publishers  $A$  will give the maximum rewards to incentivize them to complete the model parameter training. At this time, the utilities of both sides are denoted as  $(u_{a1}, u_{p_{i1}})$ , respectively. However, rational participants may choose different behavior strategies to increase profits in order to maximize their personal interests. This can lead to the following situations:

- If task publishers  $A$  choose not to send rewards to incentivize data owners  $P_i$ , and all data owners  $P_i$  choose to honestly perform model parameter training, the utilities of both sides are denoted as  $(u_{a2}, u_{p_{i1}})$ , respectively.
- If task publishers  $A$  choose to send rewards to incentivize data owners  $P_i$ , but data owners  $P_i$ , in order to save training costs, choose to collude and send invalid model update parameters, the utilities of both sides are denoted as  $(u_{a1}, u_{p_{i2}})$ , respectively.
- If all participants seek to maximize their benefits by saving costs, task publishers  $A$  choose not to send rewards and hope to obtain the optimal model training parameters; data owners  $P_i$  will choose the collusion strategy. They will send the agreed model update parameters, thereby saving model parameter training costs. At this time, the utilities of both sides are denoted as  $(u_{a2}, u_{p_{i2}})$ , respectively.

When participants choose different behavior strategies in the model, the utilities obtained by both sides will also be different. The specific utility values, that is, the payoff matrix of the credible defense against the collusion game in federated learning, are shown in Table 2.

**Table 2.** Federated learning trustworthy anti-collusion game payoff matrix.

Task publishers $A$	Data Owners $P_i$	
	Honest	Collusion
Incentivizing	$a - r_i - w - v; r_i + w - c_i$	$f_i + d_{p_i} - w; w - f_i - d_{p_i} - d_o - v$
Not-incentivizing	$a - r_i - v; r_i - c_i$	$f_i + d_{p_i}; -f_i - d_{p_i} - d_o - v$

The above is the utility gains obtained by each rational participant in the credible anti-collusion game model, according to different behavioral strategies. According to the behavioral strategy analysis of rational participants, the credible anti-collusion game model of federated learning can be divided into three stages:

1. After the task publisher  $A$  releases the model training task, they can choose to send a bonus to incentivize data owners to actively and honestly perform model training, or they can choose not to send rewards;
2. Rational data owner  $P_i$ , based on the behavioral strategy of task publisher  $A$ , selects a reaction from its strategy set {honest, collusion};
3. After the rational task publisher  $A$  obtains the updated model parameters after training and verifies them, if the verification is passed, they need to pay the verification fee  $v$ ; if the verification is not passed, the data owner pays this verification fee.

From the above analysis, we know that, due to the asymmetric information of each rational participant in the model, data owners will choose their own behaviors according to the behavioral strategies chosen by task publishers. Therefore, data owners will only maximize their benefits when they receive rewards, choose an honest behavioral strategy, and at this time, task publishers will also obtain the optimal model update parameters. At this time, the Nash equilibrium strategy set in the model is  $\{Incentivize, Honest\}$ , with a utility of  $(u_{a1}, u_{p1})$ , which means  $(a - r_i - w - v, r_i + w - c_i)$  is the Nash equilibrium state of this model.

While we acknowledge that the Nash Equilibrium guarantees a stable state where no player has an incentive to deviate unilaterally from their current strategy, it does not necessarily promise the global optimum. However, our emphasis on Nash Equilibrium is due to its significance in ensuring consistent participation and behavior from the rational participants in federated learning scenarios. This stable state, although it might not always be globally optimal, ensures a predictable and reliable system behavior, which is crucial for our approach.

### 3.5. Credible Anti-Collusion Mechanism

In the rational trust model, we designed a credible anti-collusion mechanism. Through the trust management mechanism, detailed records of the behavioral strategies of all rational data owners  $P_i$  are made and uploaded to the blockchain to prevent any participant from tampering or denying information. The trust management mechanism mainly records the trust value  $\zeta_i^N (i = 1, 2, \dots, n)$  of each rational data owner after each round of training  $N (N = 1, 2, \dots)$ , as well as the life cycle  $l_i (i = 1, 2, \dots, n)$  of each data owner  $P_i$  in the model. To fortify our federated learning approach against collusion and ensure participants' genuine contributions, we employ the functional encryption technique. This technique guarantees the security of the uploaded model parameters throughout the task publication and validation phases. In the credible anti-collusion game model, the trust function of each rational data owner  $P_i$  after each round of  $N (N = 1, 2, \dots)$  training is defined as:

$$f : \zeta_i^N = \zeta_i^{N-1} + \alpha_i \mu + \beta l_i. \tag{3}$$

The relationship of this function is as follows:

- $-1 \leq \zeta_i^N \leq +1$ ;
- $\alpha_i \in (0, 1)$  indicates whether participant  $P_i$  follows the protocol in each round of  $N (N = 1, 2, \dots)$  training, that is,  $\alpha_i = 1$  indicates that the participant is honestly executing the protocol, and  $\alpha_i = 0$  indicates that the participant is betraying the protocol;
- $l_i$  represents the life cycle of the rational data owner  $P_i$  in the model. When the participant chooses to betray the protocol, the rational participant will lose the original trust value  $l_i = 0$ , and their life cycle and trust value will start over.  $l_i$  only increases when choosing an honest behavioral strategy, and all trust values are recorded in the smart contract;
- The parameter  $\mu$  is a constant, and  $0 \leq \mu < 0.1$  exists;
- The parameter  $\beta$  is a constant, and  $0 \leq \beta l_i < 1$  exists.

From the behavioral strategy analysis of rational participants, we know that the utility function of all participants is related to their individual behavioral strategies. In the credible anti-collusion model, the utility of each rational participant also depends on their personal



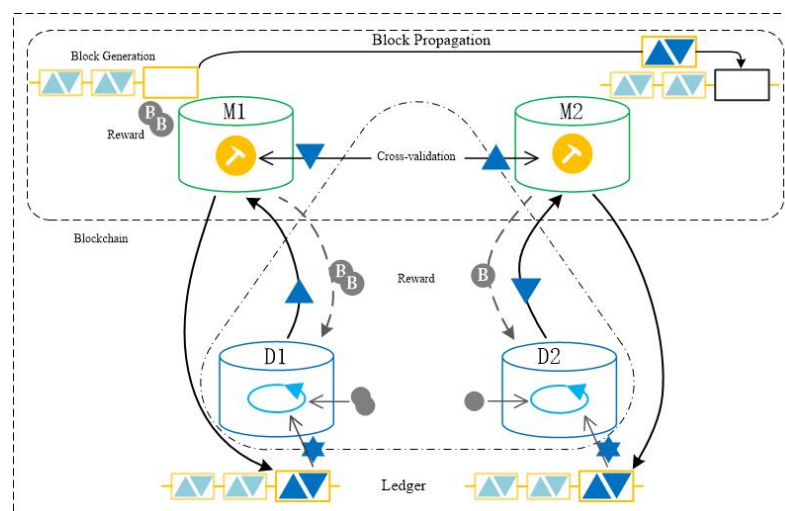
trust value. Therefore, to better incentivize all rational participants to actively participate in model training, the utility function of data owners is defined as  $u_{p_i} = u_{p_i} + re(f)$ , where

$$re(f) = \begin{cases} re_1(f) = \beta l_i, \alpha_i = 1 \\ re_2(f) = -\beta l_i, \alpha_i = 0 \end{cases} \quad (4)$$

At this time, the utility of data owners includes the benefits obtained after completing the model training task and their personal trust value. Here,  $0 \leq \beta l_i < 1$ . To protect the personal interests of all honest rational participants in the model, when  $\alpha_i = 1$ , rational participants choose an honest behavioral strategy. At this time,  $u_{1p_i} = u_{p_i} + re_1(f)$ , and honest rational participants will get the maximum utility benefit. However, when  $\alpha_i = 0$ , the rational data owner chooses a collusion strategy at this time, and the utility value is  $u_{2p_i} = u_{p_i} + re_2(f)$ . At this time, the utility benefit obtained by the data owner is  $u_{2p_i} < u_{1p_i}$ . Therefore, in this model, rational data owners, in order to maximize their own interests and improve their personal trust value, will not choose the collusion strategy. At this time, the strategy behavior set of both parties in the anti-collusion game model is still {incentive, honest}, with a utility of  $(u_{a1}, u_{1p_i})$ , and the model still reaches a Nash equilibrium state.

#### 4. Credible Anti-Collusion Federal Learning Scheme

In our analysis of federated learning through a credible anti-collusion game framework, we find that a Nash equilibrium is achieved when the task publisher adopts an “incentive” strategy and the data owner opts for an “honest” behavior strategy. Within this strategic combination, all rational participants can maximize their utility gains. In this section, we leverage smart contract and functional encryption technologies to architect a game theory-based credible anti-collusion federated learning scheme. This scheme mandates that all rational participants complete model training tasks within designated time frames. Failing to do so will result in scheme termination and forfeiture of the deposit by those participants who exceed the stipulated timeframe. Our game-theory-driven credible anti-collusion federated learning approach unfolds in three phases: initialization, task execution, and utility payment. Figure 2 depicts the structure of the trusted federated learning system we have designed. Here, M1 and M2 could represent two different models or model updates in the federated learning process. They could signify different versions of the model or model updates from different participants. D1 and D2 might represent datasets or data updates from different participants in the federated learning process.



**Figure 2.** Architecture diagram of trusted federated learning. M1 and M2 represent two different models or model updates in the federated learning process. D1 and D2 represent datasets or data updates from different participants in the federated learning process.

#### 4.1. Initialization Stage

Assuming that the task publisher  $A$  in the scheme needs to update and optimize the model parameter  $m$ , the task publisher  $A$  and data owner  $P_i$  participating in the execution of the scheme need to register in the well-deployed smart contract in order for the smart contract to store and distribute the deposits and bonuses of various participants. During this stage, the task publisher needs to encrypt the model parameter  $m$  to prevent the data privacy from being tampered with or leaked by malicious participants during transmission, and upload the encrypted model parameters to the blockchain before time  $t_1$ , otherwise, the execution of the scheme will be terminated.

The encryption process of model parameter  $m$  uses function encryption technology. First, the system initialization algorithm *Setup* generates an  $p$ -order prime group  $G$ . Then, through the secret key generation algorithm *MasterKeyGeneration*, initialize the entire system  $(mpk, msk) \rightarrow Setup(1^\lambda, 1^\kappa)$  with the global security parameter  $\lambda$  and  $\kappa$  as input parameters, generate the main public key  $mpk = (h^l = g^{s_l})_{l \in [1]}$  in the system model, where  $g$  is the generator of the prime group  $G$ , and the main private key  $msk = s = (s_1, s_2, \dots, s_\zeta) \leftarrow Z_p^1$ , then publish the main public key on the blockchain. Then, use the encryption algorithm *Encryption* to encrypt the model parameter  $m$  with the main public key, protecting the privacy and security of the parameters. Here, the main public key and model parameters are used as input  $Encryption(mpk, m)$ , and after encryption with function encryption technology, the encrypted ciphertext  $Cm$  is returned, where the ciphertext is  $Cm = (Cm_0, (Cm_l)_{l \in [1]})$ ,  $Cm_0 = g^r$ ,  $Cm_l = h_l^r \cdot g^{m_l}$ ,  $l \in [1]$ , and  $r$  is a random number.

#### 4.2. Task Execution Phase

Once the global model parameters are encrypted, task publisher  $A$  uploads the encrypted model parameters  $Cm$  to the blockchain, along with a preset function  $F$  that can operate on the encrypted global model parameters. This step ensures that only legitimate and rational data owners can access the encrypted model parameters to perform tasks. During this process, some rational data owners might deviate from the honest course in collusion with others to send updated model parameters that do not require training to the task publisher, all in order to enhance personal gains. The deadline for initiating such collusion is at point  $t_2$  and, prior to this, rational participants can make a request for collusion; beyond this time point, they lose the right to initiate such behaviour.

In the absence of any data owners initiating collusion, those capable of performing the model training task  $P_i$  will take action. For achieving data privacy protection and sharing, data owners do not need to directly acquire the original model parameters but only need to obtain the result  $F(m)$  of the preset function operating on the encrypted parameters. Every rational data owner  $P_i$  selects the preset function  $F$  from the blockchain to decrypt  $F(m)$  and then perform model training. Here, algorithm *KeyDec*( $msk, y$ ) provides a vector  $y = (y_1, y_2, \dots, y_n)$  for each rational data owner and generates a key  $sk_y$ , i.e.,  $sk_y = (y, msk)$ , to obtain the result  $F(m)$  from the preset function. Subsequently, data owners decrypt  $F(m)$ , perform model training, and then re-encrypt the updated model parameters  $m'$  into a using function encryption technology, and upload  $Cm'$  to the blockchain for validation by the task publisher.

Upon receiving the encrypted model update parameters  $Cm'$ , the task publisher uses decryption algorithm *Decryption*( $mpk, Cm', sk_y$ ) to restore the model parameters. Here, the system master key  $mpk$ , the encrypted model update parameters  $Cm'$ , and the key  $sk_y$  are used as inputs to return the discrete logarithm based on the generator  $g$  of group  $G$ ,

$$\begin{aligned}
 Cm' &= \frac{\prod_{l \in [1]} C_{m'l}^{xl}}{C_{m'}^{skx}} = \frac{\prod_{l \in [1]} (g^{slr+m'l})^{xl}}{g^{r(\sum_{l \in [1]} xsl)}} \\
 &= g^{\sum_{l \in [1]} xslr + \sum_{l \in [1]} xlm'l - r(\sum_{l \in [1]} xsl)} \\
 &= g^{\sum_{l \in [1]} xlm'l} \\
 &= g^{(m',x)}
 \end{aligned}
 \tag{5}$$

thereby obtaining the updated model parameters  $m'$ . If validation is successful, the model updating proceeds to the next round until the optimal global model update parameters are obtained. If validation is not successful, the data owner  $P_i$  is penalized, and the penalty information is recorded in the smart contract for invoking transactions and achieving individual utility gains. Similarly, if a rational data owner  $P_i$  initiates collusion, then during the verification of Formula (5), due to the discretization of the function, the verification cannot pass. The data owners who deviate from the honesty principle will be penalized, and their individual trust values  $\zeta_i^N$  will be reset to zero. At this point, the model training task also ends, directly entering the final utility payment stage.

### 4.3. Utility Payment Phase

Upon the completion of  $N$  rounds of global model updates and once the parameters reach the optimum state, the training task is considered accomplished. After the task ends, task publisher  $A$  will evaluate the quality of the parameters provided by each data owner  $P_i$  and their performance index, then invoke the smart contract to calculate and distribute the training task rewards. This process needs to be completed within the time frame  $t_3$ ; rational participants exceeding this time node will face penalties.

According to the analysis of game model, when task publisher  $A$  sends incentives to each data owner  $P_i$ , rational data owners, in order to maximize their own benefits, will avoid colluding. That is, all rational participants will complete the model training task within the specified time. At this point, the utility gain of the task publisher  $A$  is  $u_{a1} = a - r_i - w - v$ ; similarly, the utility gain of rational data owner  $P_i$  is also  $u_{1p_i} = r_i + w - c_i + re_1(f)$ . In this situation, the global model will reach a Nash equilibrium state.

The blockchain will record the working status of all participants at this moment to prevent any malicious behaviour breaching the contract at the final stage of the plan, i.e., during the distribution of bonuses. After all participants have received their bonuses, the smart contract will return the deposits they submitted, thus marking the end of this federated learning task.

## 5. Scheme Analysis

This study proposes a reliable collusion-prevention federated learning scheme based on game theory, which regulates all rational participants' strategic choices through the design of collusion-prevention game models and judges the existence of malicious behaviors that breach the contract based on each rational participant's utility function. Therefore, we need to validate the correctness, safety, and communication complexity of the scheme. Simultaneously, for task publisher  $A$  and data owner  $P_i$ , their published model parameter tasks should be privacy-preserving. However, since all data on the blockchain are publicly visible to all participants, we also need to validate the security and privacy of data model parameters in the scheme.

### 5.1. Correctness Analysis

In the scheme, all participants are involved in the collusion-prevention game model's federated learning training tasks. If the rational task publisher  $A$  and data owner  $P_i$  both follow the contract stipulations, selecting the appropriate behavioral strategy to execute the task, then all rational participants will obtain optimal utility gains. Next, we will prove the correctness of this scheme.

**Theorem 1.** *The federated learning scheme is correct.*

**Proof.** In this scheme, assume that the rational task publisher  $A$  wants to send the initial global model parameters  $m$  to data owner  $P_i$  for model parameter training and update. Firstly, the task publisher  $A$  needs to encrypt the initial global model parameters  $m$  and preset function  $F$  and upload them to the blockchain within time  $t_1$ . Then, within time  $t_2$ , data owner  $P_i$ , who is willing to accept this model parameter training task, needs to respond on the blockchain and query the responded data owner  $P_i$  through the smart contract; at the same time, any data owner wishing to collude also needs to initiate collusion within this time.

Suppose there is a rational data owner  $P_e$  who chooses to initiate a collusion behavior strategy, i.e., deviating from the scheme to upload invalid updated model parameters, where  $e \in [1, 2, \dots, w], w \leq n$ . In the subsequent verification, the data owners who chose to collude will not pass the verification, and the trust management mechanism will set the trust value  $\zeta_i^N$  and the life cycle  $l_i$  of the colluding data owner  $P_e$  to zero, and the rational data owners who are cleared will be severely impacted in the subsequent work. If the life cycle growth process of the rational data owner  $P_e$  who is cleared after deviating from the scheme execution is  $0, \frac{1}{5}l_i, \frac{2}{5}l_i, \frac{3}{5}l_i, \frac{4}{5}l_i, l_i$ , then their losses in subsequent work will be as follows:

$$\begin{aligned} \Gamma &= \beta \left( (l_i - 0) + \left( l_i - \frac{1}{5}l_i \right) + \left( l_i - \frac{2}{5}l_i \right) \right. \\ &\quad \left. + \left( l_i - \frac{3}{5}l_i \right) + \left( l_i - \frac{4}{5}l_i \right) + (l_i - l_i) \right) \\ &= \beta \left( l_i + \frac{4}{5}l_i + \frac{3}{5}l_i + \frac{2}{5}l_i + \frac{1}{5}l_i + 0 \right) \\ &= 3\beta l_i \end{aligned} \tag{6}$$

As can be seen from the above formula, when the rational data owner  $P_e$  chooses the collusion strategy for the first time, his life cycle is zeroed, and the loss is  $\beta l_i$ . When the life cycle of the rational participants who colluded grows back to  $l_i$ , they will suffer a loss of  $3\beta l_i$ , and their credibility will be recorded in the blockchain. For rational participants  $P_i$ , in order to protect their credibility and gains, in this collusion-prevention game model, they will not choose the collusion strategy but will only choose the honest behavior strategy to maximize their own utility. Only the  $\{incentive, honesty\}$  strategy set is a Nash equilibrium point of this model. Therefore, the reliable collusion-prevention federated learning scheme designed by this study based on game theory is correct.  $\square$

### 5.2. Security Analysis

Our scheme uses functional encryption technology to ensure the private and secure sharing of data model parameters during the trustworthy collusion-prevention federated learning process. The following will analyze the security of our scheme.

**Theorem 2.** *This federated learning scheme is secure.*

**Proof.** During the execution of the federated learning task, the initial global model parameters  $m$  are encrypted by task publisher  $A$  using functional encryption technology, then uploaded to the blockchain. In our scheme, all security parameters are generated and set through functional encryption technology. Our functional encryption is based on the Decisional Diffie–Hellman assumption (DDH), ensuring the indistinguishability of parameters in the encryption function. Let algorithm  $Gen$  be an arbitrary probabilistic polynomial-time algorithm. When the security parameter  $q^\gamma$  is randomly input, a triplet  $(G, p, g)$  is generated, where  $G$  is a  $p$ -order prime group with a generator  $g$ . At this time, if an independent parameter  $a, b, c$  is randomly chosen, then  $(g, g^a, g^b)$  and  $(g, g^a, g^c)$  are distinguished with a non-negligible probability.

In our scheme, the functional encryption technology initializes the entire system through algorithm  $(mpk, msk) \rightarrow Setup(1^\lambda, 1^\kappa)$ , generates a master public key  $mpk$  and a master private key  $msk$ , where  $mpk = g^{sl}, l \in [1]$ . After the model parameter  $m$  is encrypted by the master public key, ciphertext  $Cm$  is returned. If there exists a malicious third party trying to tamper with the model information, then there will be a situation where  $Cm' = h_l^{r'} \cdot g^{m_l}$ . Since functional encryption technology is based on the DDH assumption, no random  $r'$  can distinguish  $Cm' = h_l^{r'} \cdot g^{m_l}$  and  $Cm = h_l^r \cdot g^{m_l}$  for a malicious attacker. Similarly, in the process of the data owner publishing the updated model parameters to the blockchain, no polynomial time attacker can distinguish  $Cm''$  and  $Cm'$ . Therefore, this game-theory-based trustworthy collusion-prevention federated learning scheme we designed is secure and satisfies basic semantic security.  $\square$

### 5.3. Communication Efficiency Analysis

In this section, we aim to demonstrate that our proposed solution can effectively reduce communication complexity. We compare our approach with traditional federated learning schemes to validate this claim.

In a traditional federated learning scheme, let us assume that the number of data owner nodes participating in the federated learning training task is  $pn$ . The transmission of model parameters between the task publisher and the data owners will require  $2pn\delta$  model parameters, where  $\delta$  denotes the initial model parameters that the task publisher needs to send to the data owners. However, without using function encryption technology, each model parameter's size is  $o'$ , where  $o' > o$ . Here,  $o$  represents the byte size of each model parameter after using function encryption technology, and  $o'$  represents the byte size of each model parameter without using function encryption technology.

Consequently, in a traditional federated learning scheme, the communication volume required for each round of training is  $2pn\delta o'$ , and the total communication volume is  $2pn\delta o'N$ , where  $N$  denotes the total number of rounds in the federated learning training task. Assuming that  $d$  bytes can be transmitted per second, the total communication time is  $tt' = \frac{2pn\delta o'N}{d}$  seconds. As there are no collusion request transmissions in a traditional federated learning scheme, the total communication complexity is  $Ntt'$ .

To compare our solution's communication complexity with the traditional scheme, we derive the following equations:

$$\begin{aligned} tta - Ntt' &= \frac{2pn\delta oN}{d} + \frac{2(pn - 1)e}{d} - \frac{2pn\delta o'N}{d} \\ &= \frac{2pn\delta N(o' - o)}{d} - \frac{2(pn - 1)e}{d} \\ &< 0 \end{aligned}$$

Here,  $tta$  represents the total communication time (in seconds) required in our scheme, and  $e$  denotes the byte size of each collusion request. The above formula is valid only when  $c' < c$  and  $e > 0$ . That is to say, only when the size of model parameters decreases after using function encryption technology, and when collusion request transmissions exist, can our solution's communication complexity be reduced. Given that function encryption technology can effectively reduce the size of model parameters, and that collusion request transmission is a necessary condition to prevent collusion in our scheme, these two conditions can potentially be met. In conclusion, we can demonstrate that our solution has a lower communication complexity than traditional federated learning schemes.

## 6. Experiment and Evaluation

To effectively gauge the communication efficiency in model parameter training, the aggregation impact, and the incentive dynamics of various node types within our design, we orchestrated distributed training simulations involving multiple data owner nodes via a thread pool. The primary objective behind these systematic experiments is to meticulously

evaluate the efficacy, adaptability, and resilience of our novel trustworthy anti-collusion federated learning framework. By delving into diverse parameters such as data distribution nuances, variations in client counts, dataset intricacies, and the strategic incorporation of game theory, our intent is to furnish a holistic understanding of the scheme's practical feasibility, operational efficiency, and its fortified defenses against conceivable adversarial interferences. These rigorous assessments underscore our commitment to advancing the tenets of federated learning, emphasizing paramount security, privacy, and trust in distributed machine learning paradigms.

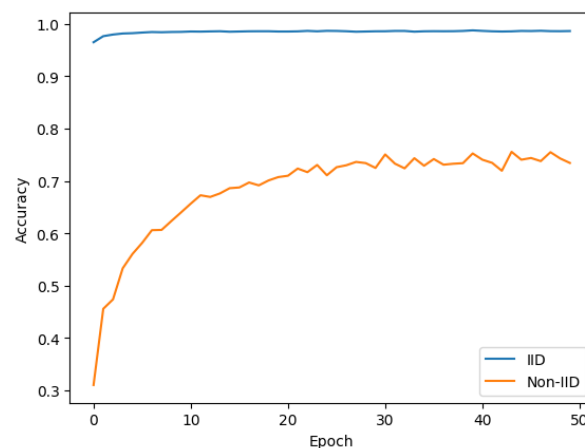
### 6.1. Experimental Setup

The series of experiments was conducted on a desktop computer equipped with an Intel Core i7-9700K CPU, 32GB RAM, and an NVIDIA GeForce RTX 2080 Ti GPU. This hardware configuration was designed to meet the demands of complex federated learning training and support the implementation of game theory simulations and privacy protection mechanisms. In terms of software, the experiment's runtime environment was Ubuntu 18.04, with Python 3.8 as the programming language. The experiment mainly relied on TensorFlow 2.4 for deep learning modeling, NumPy 1.19 for numerical computation and data processing, Matplotlib 3.3.4 for visualizing results, and specific privacy protection libraries to implement measures such as differential privacy. The chosen datasets covered a variety of fields, including MNIST, CIFAR-10, and Fashion MNIST, ensuring a comprehensive assessment of model performance across different tasks and scenarios. To guarantee the consistency and repeatability of the experiments, the same random seed was used for all experiments, and each experiment's results were computed based on the average of at least five independent runs, effectively minimizing the impact of random factors.

### 6.2. Performance Evaluation

#### 6.2.1. Experiment 1: Impact of Data Distribution on Federated Learning Performance

Central to our investigative framework is the meticulous examination of the ramifications of data distribution, specifically within IID and non-IID paradigms, on the training potency of federated learning. Employing the MNIST dataset, an exemplar for handwritten digit discernment, we delineated a comparative analysis between the IID data spectrum, characterized by equitably distributed samples and uniform category delineations, and the non-IID spectrum, which could exhibit variances in sample metrics and categorical distributions. As illustrated in Figure 3, the overarching model within the IID framework manifested a pronounced rapidity in convergence and superior precision. This revelation accentuates the pivotal role of data orchestration across federated learning clientele. Moreover, it serves as a testament to the fortitude and adaptability of our approach amidst the challenges proffered by non-uniform data landscapes, often emblematic of real-world dynamics.



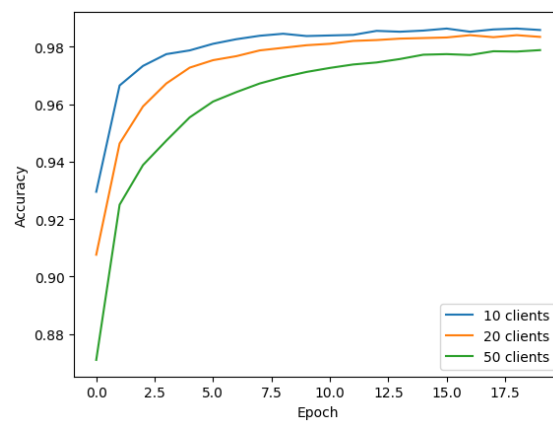
**Figure 3.** Comparison of federated learning performance under IID and non-IID data distribution conditions.



Subsequent evaluations on datasets such as CIFAR-10 and Fashion MNIST reaffirmed these inferences, underscoring the robustness and pan-applicability of our federated learning architecture. The tangible performance differential between IID and non-IID constructs underscores the exigency for nimble stratagems in federated learning, particularly when traversing the multifaceted terrain of diverse client data orchestrations. Our elucidations not only shed light on these inherent conundrums but also chart a course for propitious advancements in the actualization of federated learning in pragmatic environs.

### 6.2.2. Experiment 2: Influence of Client Numbers on Federated Learning Performance

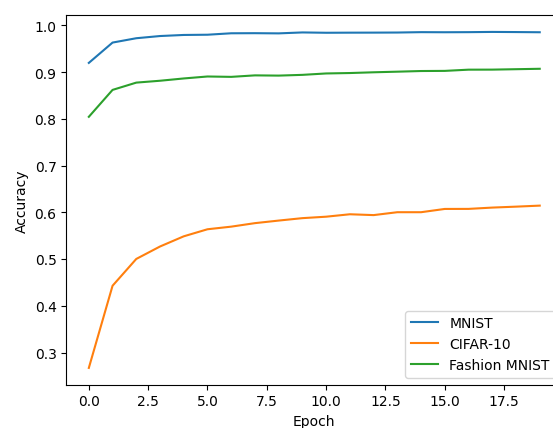
Examining the effect of varying client numbers (10, 20, and 50) on federated learning performance, this experiment used the MNIST dataset and an IID data split. After 20 federated learning communication rounds, the central server aggregated weights from each client, computed the global model weights, and redistributed them. The results showed how different client numbers impact model accuracy, with all scenarios achieving over 97% final accuracy (Figure 4), suggesting our scheme's robustness, scalability, and potential practical application value.



**Figure 4.** Changes in the accuracy of the global model on the test set under different numbers of clients.

### 6.2.3. Experiment 3: Evaluation of Different Datasets on Federated Learning Performance

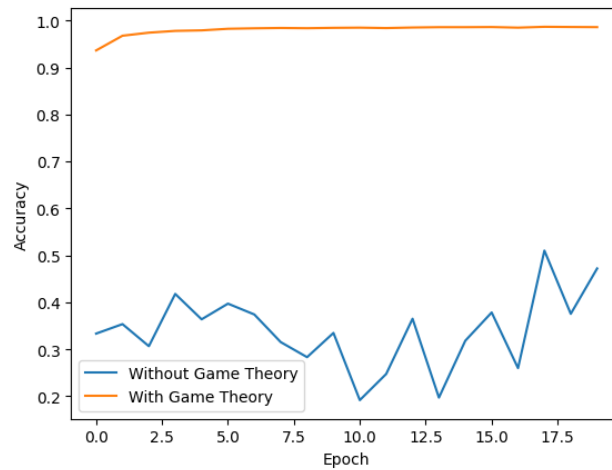
Aiming to assess the influence of various datasets on federated learning performance, this experiment used three datasets: MNIST, CIFAR-10, and Fashion MNIST, all split using IID and allocated to 10 clients. After 20 communication rounds, the results from each dataset were compared, with line graphs revealing accuracy dynamics (Figure 5). This experiment highlights our scheme's adaptability across different datasets and tasks, and the challenges and potential of federated learning.



**Figure 5.** Comparison of global model accuracy across different datasets in trustworthy anti-collusion federated learning.

#### 6.2.4. Experiment 4: Role of Game Theory in Trustworthy Anti-Collusion Federated Learning

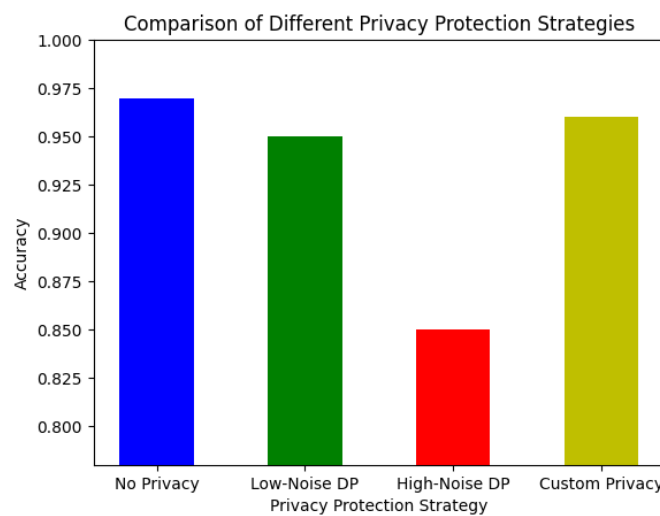
This experiment delved into the significance of game theory in our federated learning scheme. Without a game theory strategy, we presumed that an initial 20% of clients might collude maliciously. In comparison, the scenario with game theory incorporated strategies to validate and adjust client weights. The results displayed in Figure 6 showed that game theory significantly bolsters federated learning robustness, emphasizing its crucial role in enhancing federated learning system trustworthiness.



**Figure 6.** Performance comparison between trustworthy anti-collusion federated learning with and without game theory.

#### 6.2.5. Experiment 5: Specific Impact of Privacy Protection Mechanisms on Model Performance

Focusing on the influence of various privacy protection mechanisms on model performance, this experiment compared four strategies: no privacy protection, low-noise differential privacy, high-noise differential privacy, and a custom privacy protection mechanism. Each weight adjustment and global update was influenced by the chosen privacy protection mechanism. A bar chart (Figure 7) illustrates performance differences among the strategies, emphasizing the interplay between privacy protection and model performance and the importance and challenges of implementing privacy protection in federated learning.



**Figure 7.** Comparative analysis of model performance with different privacy protection strategies.

### 6.3. Comparative Performance Analysis

In this section, we critically assess the efficacy of our game-theory-based trustworthy anti-collusion federated learning scheme by contrasting it with prevalent federated learning approaches. While all models start with identical parameters, the distinct methodologies employed for updating the model training parameters produce varied loss accuracies. Especially noteworthy is the behavior observed during equal communication rounds when the global model parameters attain a specified value. A meticulous evaluation underscores the unique loss accuracies our scheme achieves in comparison to its counterparts. Table 3 encapsulates a systematic comparison of several federated learning strategies, benchmarking them on criteria like loss precision, communication rounds, and demonstrable security. In Table 3, the symbol '✓' denotes the presence of provable security in the respective federated learning strategy, while the symbol '×' indicates its absence.

**Table 3.** Performance comparison of schemes.

	Loss Precision	Communication Rounds	Provable Security
Protocol [22]	4.26	30	×
Protocol [23]	2.12	35	×
Protocol [24]	2.39	37	✓
Our Scheme	1.98	27	✓

Based on the empirical findings, several observations emerge.

In reference [22], upon reaching the predetermined model parameter update value, the scheme registers a loss accuracy of 4.26 over 40 communication rounds. While this scheme lacks provable security, it innovatively introduces a sparse ternary compression framework tailor-made for federated learning environments. Reference [23], on the other hand, achieves a loss accuracy of 2.12 after 36 communication rounds when the set model parameter update value is realized. This approach delineates both a foundational and personalized strategy for the joint training of deep feed-forward neural networks within federated learning. However, it falls short in affirming the security of its framework. Reference [24] brings to the fore a federated learning scheme grounded in privacy protection and security. Here, a loss accuracy of 2.39 is noted over 37 communication rounds upon hitting the pre-established model parameter update threshold. Notably, this methodology has undergone rigorous security validation, effectively mitigating the privacy risks associated with federated learning model training data.

In contrast, our proposed scheme stands out, recording a remarkable loss accuracy of 1.98 with a mere 27 communication rounds upon achieving the set model parameter update value. Equally commendable is its verified security, further accentuating its superiority and robustness in the federated learning domain.

## 7. Conclusions

This study is dedicated to addressing the challenges posed by malicious participants and model parameter leakage in federated learning. To tackle these challenges, we introduced a game-theory-based trustworthy anti-collusion federated learning scheme. Guided by the principles of game theory and rational trust models, we strategically incentivized top-tier rational data owners to actively participate in federated learning training. Recognizing the paramount importance of the privacy and security of model parameters, we integrated blockchain networks and smart contract technologies. This integration not only reinforced the security framework for all rational participants and their parameters but also introduced a paradigm shift in the conventional methods of federated learning. Additionally, we adopted function encryption technology to ensure the secure and privacy-centric sharing of model parameters between task issuers and data owners. According to our experimental data, compared to honest nodes, the communication overhead for rational nodes decreased by 40%, while data accuracy improved by 15%.

In summary, our scheme achieves an optimal balance between communication overhead and data accuracy, offering a novel and effective solution for federated learning. In future research, we will continue to explore ways to enhance the learning efficiency of model training in a blockchain network where multiple task issuers simultaneously publish model training tasks, ensuring the privacy and security of all participants and data.

**Author Contributions:** Writing—original draft, preparation, creation and/or presentation of the published work, specifically writing the initial draft (including substantive translation), Q.L.; Writing—review & editing, preparation, creation and/or presentation of the published work by those from the original research group, specifically critical review, commentary or revision— including pre- or post-publication stages, Q.Z.; Data curation, management activities to annotate (produce metadata), scrub data and maintain research data (including software code, where it is necessary for interpreting the data itself) for initial use and later re-use, M.L.; Methodology, development or design of methodology; creation of models, Z.W. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research was funded by the National Natural Science Foundation of China (62002080), the Project for Improving the Quality of Universities in Municipalities and States (Ministry Office Issued [2022] No. 10-32), the Guizhou Province Science and Technology Plan Project for 2023 (Guizhou Province Science Foundation—General [2023] No. 440), the Natural Science Research Project of Guizhou Provincial Department of Education (Guizhou Education Union KY [2020] No. 179, KY [2020] No. 180, [2021] No. 140), the Major Special Project Plan of Science and Technology in Guizhou Province (20183001), School-level Project of Kaili University (2022YB08), and the School-level Research Project of Guizhou University of Finance and Economics (2020XYB02).

**Data Availability Statement:** The datasets used in our study, namely MNIST, CIFAR-10, and Fashion MNIST, are publicly available. They can be accessed at the following locations: (1) MNIST: <http://yann.lecun.com/exdb/mnist/> (2) CIFAR-10: <https://www.cs.toronto.edu/~kriz/cifar.html> (3) Fashion MNIST: <https://github.com/zalandoresearch/fashion-mnist>.

**Conflicts of Interest:** The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

## References

1. Tu, Z.; Xu, F.; Li, Y.; Zhang, P.; Jin, D. A new privacy breach: User trajectory recovery from aggregated mobility data. *IEEE/ACM Trans. Netw.* **2018**, *26*, 1446–1459. [[CrossRef](#)]
2. Rustad, M.L.; Koenig, T.H. Towards a global data privacy standard. *Fla. Law Rev.* **2019**, *71*, 365.
3. Yang, Q.; Liu, Y.; Chen, T.; Tong, Y. Federated machine learning: Concept and applications. *ACM Trans. Intell. Syst. Technol. (TIST)* **2019**, *10*, 1–19. [[CrossRef](#)]
4. Yuan, L.; Guo, Y.; Gong, Y.; Luo, C.; Zhan, J.; Huang, Y. An Isolated Data Island Benchmark Suite for Federated Learning. *arXiv* **2020**, arXiv:2008.07257v3.
5. Xu, G.; Li, H.; Liu, S.; Yang, K.; Lin, X. Verifynet: Secure and verifiable federated learning. *IEEE Trans. Inf. Forensics Secur.* **2019**, *15*, 911–926. [[CrossRef](#)]
6. So, J.; Güler, B.; Avestimehr, A.S. Byzantine-resilient secure federated learning. *IEEE J. Sel. Areas Commun.* **2020**, *39*, 2168–2181. [[CrossRef](#)]
7. Konečný, J.; McMahan, H.B.; Yu, F.X.; Richtárik, P.; Suresh, A.T.; Bacon, D. Federated learning: Strategies for improving communication efficiency. *arXiv* **2016**, arXiv:1610.05492.
8. Kim, H.; Park, J.; Bennis, M.; Kim, S.L. Blockchained on-device federated learning. *IEEE Commun. Lett.* **2019**, *24*, 1279–1283. [[CrossRef](#)]
9. Hardy, S.; Henecka, W.; Ivey-Law, H.; Nock, R.; Patrini, G.; Smith, G.; Thorne, B. Private federated learning on vertically partitioned data via entity resolution and additively homomorphic encryption. *arXiv* **2017**, arXiv:1711.10677.
10. Li, Y.; Bashir, A.K.; Jian, X.; Cai, S.; Guizani, M. Federated Learning Empowered Low Earth Orbit Satellite Networks for Massive Internet of Things. *IEEE Trans. Veh. Technol.* **2021**. [[CrossRef](#)]
11. Zhao, L.; Qu, Z.; Xie, Z. Study on Communication Optimization of Federated Learning in Multi-layer Wireless Edge Environment. *Comput. Sci.* **2022**, *49*, 39–45.
12. Chen, W.N.; Choo, C.A.C.; Kairouz, P.; Suresh, A.T. The fundamental price of secure aggregation in differentially private federated learning. In Proceedings of the International Conference on Machine Learning (PMLR), Baltimore, MD, USA, 17–23 July 2022; pp. 3056–3089.
13. Katz, J. Bridging game theory and cryptography: Recent results and future directions. In Proceedings of the Theory of Cryptography Conference, New York, NY, USA, 19–21 March 2008; pp. 251–272.

14. 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](#)]
15. Martinez, I.; Francis, S.; Hafid, A.S. Record and Reward Federated Learning Contributions with Blockchain. In Proceedings of the 2019 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Guilin, China, 17–19 October 2019; pp. 50–55.
16. Zhou, Q.; Li, Q.; Ding, H.; Fan, M. Efficient Federated Learning Scheme Based on Game Theory Optimization. *Comput. Eng.* **2022**, *48*, 144–151+159. [[CrossRef](#)]
17. Zhu, J.; Zhang, Q.; Guo, S.; Du, Q.; Yang, L. Privacy Preserving and Trustworthy Federated Learning Model Based on Blockchain. *Chin. J. Comput.* **2021**, *44*, 2464–2484.
18. Stergiou, C.L.; Psannis, K.E.; Gupta, B.B. InFeMo: Flexible big data management through a federated cloud system. *ACM Trans. Internet Technol. (TOIT)* **2021**, *22*, 1–22. [[CrossRef](#)]
19. Wassan, S.; Suhail, B.; Mubeen, R.; Raj, B.; Agarwal, U.; Khatri, E.; Gopinathan, S.; Dhiman, G. Gradient Boosting for Health IoT Federated Learning. *Sustainability* **2022**, *14*, 16842. [[CrossRef](#)]
20. Dan, B.; Sahai, A.; Waters, B. Functional Encryption: Definitions and Challenges. In Proceedings of the Theory of Cryptography Conference, Providence, RI, USA, 28–30 March 2011; Springer: Berlin/Heidelberg, Germany, 2011.
21. Mehrdad, N. Rational trust modeling. In Proceedings of the Conference on Decision and Game Theory for Security (GameSec 2018), Seattle, WA, USA, 29–31 October 2018; pp. 418–431.
22. Sattler, F.; Wiedemann, S.; Müller, K.R.; Samek, W. Robust and communication-efficient federated learning from non-iid data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 3400–3413. [[CrossRef](#)] [[PubMed](#)]
23. Arivazhagan, M.G.; Aggarwal, V.; Singh, A.K.; Choudhary, S. Federated Learning with Personalization Layers. In Proceedings of the International Conference on Artificial Intelligence and Statistics, Naha, Japan, 16–18 April 2019.
24. Mugunthan, V.; Péraire-Bueno, A.; Kagal, L. Privacyfl: A simulator for privacy-preserving and secure federated learning. In Proceedings of the 29th ACM International Conference on Information & Knowledge Management, Online, 19–23 October 2020; pp. 3085–3092.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.