Hindawi
Mobile Information Systems
Volume 2021, Article ID 6633332, 18 pages
https://doi.org/10.1155/2021/6633332

*Research Article*

# BFLP: An Adaptive Federated Learning Framework for Internet of Vehicles

**Yongqiang Peng, Zongyao Chen, Zexuan Chen, Wei Ou, Wenbao Han, and Jianqiang Ma**

*School of Computer Science and Cyberspace Security, Hainan University, Haikou 570228, Hainan, China*

Correspondence should be addressed to Wei Ou; ouwei@hainanu.edu.cn

Received 28 December 2020; Revised 12 February 2021; Accepted 16 February 2021; Published 2 March 2021

Academic Editor: Xiaoxian Yang



Applications of Internet of Vehicles (IoV) make the life of human beings more intelligent and convenient. However, in the present, there are some problems in IoV, such as data silos and poor privacy preservation. To address the challenges in IoV, we propose a blockchain-based federated learning pool (BFLP) framework. BFLP allows the models to be trained without sharing raw data, and it can choose the most suitable federated learning method according to actual application scenarios. Considering the poor computing power of vehicle systems, we construct a lightweight encryption algorithm called CPC to protect privacy. To verify the proposed framework, we conducted experiments in obstacle-avoiding and traffic forecast scenarios. The results show that the proposed framework can effectively protect the user's privacy, and it is more stable and efficient compared with traditional machine learning technique. Also, we compare the CPC algorithm with other encryption algorithms. And the results show that its calculation cost is much lower compared to other symmetric encryption algorithms.

## 1. Introduction

Internet of Vehicles (IoV) is a new type of industry with deep integration of automobile, electronics, information communication, transportation, and traffic management. As a necessary technical means to realize the intelligent transportation system and automatic driving, IoV is the core technology to solve the current traffic problems. In IoV, data exchange between on-board unit (OBU), roadside unit (RSU), and mobile network enable information sharing between vehicles and all systems. For example, in vehicle-to-vehicle (V2V) communications [1], which is one of the IoT applications, the functions running on our sensor nodes can be part of the on-board system of each vehicle and the functions running on base stations can be running on roadside devices. In this way, the local transportation department can accurately and comprehensively grasp the real-time traffic information to conduct intelligent analysis and make decisions. And the current decision information is fed back to each vehicle for emergency warning of traffic incidents and path planning [2]. The development of IoV and service capability has spawned many applications, such as driving safety applications, traffic efficiency applications, and entertainment applications. At present, all countries in the world are actively researching the related technology of IoV and giving strong support to the development of it. In 2016, the U.S. Department of Transportation officially released Federal Motor Vehicle Safety Standard No. 150 (FMVSS No. 150), which mandatorily requires all light vehicles to install vehicle communication equipment to ensure the real-time transmission of safety information between vehicles. In 2018, the standardization of global unified vehicle networking communication standard LTE-V2X was completed, which is formulated by the international standard organization 3GPP. In 2020, China's national standard system for IoV is completed. With the application of IoV, vehicles can get more information than a single vehicle, which is of great significance to improve traffic efficiency, reduce the incidence of traffic accidents, and improve traffic management.

The transmission and sharing of data in IoV bring great value, but if the data are leaked in the process of

transmission, storage, and sharing, it may cause serious risks to users and society. Compared with the server, the computing power of the on-board system is usually poor, so there may be loopholes in various security protection measures. Due to the working mechanism of IoV, vehicles need to exchange BSM messages with other vehicles and RSU regularly. BSM messages are generally sent in broadcast form without the process of encryption, which brings a privacy leakage risk to each node in IoV [3]. The collection and analysis of data can help the transportation department make better decisions, but it may cause privacy problems for users. Research has shown that activities in physical space and virtual space can influence one another [4]. There have been several IoV security incidents since 2015. The vehicles are invaded remotely by attackers because of the vulnerability of the BMW connected-drive system, putting more than 2 million vehicles at the security risk of being attacked by hackers. In 2017, Nissan announced that the database of its vehicle financing department was invaded, as many as 113 million customers' personal information was hacked. In 2020, Tesla was exposed to iBeacon privacy leaks, which caused great distress to car owners' privacy.

To solve the dilemma of data silos and data privacy, Google and WeBank have proposed different federated learning (FL) algorithm frameworks. In federated learning, participants' data are kept locally, no privacy is disclosed, and participants build the federated model and benefits together [5]. Compared with traditional machine learning, the nodes in federated learning are unstable and highly autonomous. Vehicles can move at high speed and frequently and drive in and out at any time, and the density of the vehicle varies greatly in different space-time scenarios. Therefore, we can apply federated learning to IoV. Through federated learning, each vehicle can jointly train the machine learning model without exchanging local data [6].

To solve a series of security problems in IoV environment, we propose a blockchain-based federated learning pool (BFLP) framework and integrate a lightweight encryption algorithm into it. In this framework, we combine blockchain and federated learning to protect user's data privacy. Considering the poor computing power of the vehicle system, we use a lightweight encryption algorithm to encrypt the user's data to ensure security when vehicles share information. At the same time, we construct a federated learning pool module (an adaptive learning model) so that the server can select the corresponding federated learning method automatically according to different distribution characteristics of the data source. Through BFLP, user's data privacy can not only be better protected but also the model trained can be better. Our main contributions in this paper are as follows:

(i) We proposed a federated learning pool (FLP) module to train the model, which can choose the most suitable federated learning method according to the present application scenarios.

(ii) We construct a lightweight algorithm called CPC for the vehicle system to encrypt user's data.

(iii) We use the blockchain as the bottom layer of the framework to facilitate the sharing and storage of user's information, ensure the reliability of data transmission, and protect user's privacy.

The remainder of this paper is organized as follows. In Section 2, we will introduce our related work. In Section 3, we will introduce our privacy protection model in detail. In Section 4, we verify the proposed framework by experiments and discuss the experimental results and the security of our framework. Finally, Section 5 concludes this article and discusses future work.

## 2. Related Work

*2.1. Federated Learning.* The great success of AlphaGo in 2016 shows us the great potential of artificial intelligence (AI). Although AI has high commercial value, the reality is disappointing. AI depends on the data of many fields to train, but in reality, due to industry competition, privacy security, and other issues, data exist in the form of data silos. On the contrary, data security and privacy have received unprecedented attention in recent years. China implemented the Network Security Law of the People's Republic of China in 2017, and the EU formally implemented the General Data Protection Column in 2018. To solve data silos and privacy protection, federated learning was created. Figure 1 shows a traditional FL model. In federated learning, each node jointly trains the model under the coordination of the server, and the training data are stored locally without sharing with others.

Compared with traditional machine learning, federated learning does not need to collect all data but has a similar modelling effect, which greatly reduces the privacy risk and cost of machine learning. Federated learning is applied in many fields with its privacy protection. In [7], Yan et al. applied federated learning to variant perceptual learning of multisource decentralized medical image data and constructed small medical image data sets of different institutions into large medical image data sets. Kang et al. integrated federated learning modules into mobile networks to enable mobile devices to train and share models without leaking their local data [8]. Niknam et al. applied federated learning to wireless communication to protect privacy [9].

For different data sets, Yang et al. [5] divided federated learning into horizontal federated learning, vertical federated learning, and federated transfer learning, which enriched the scope of federated learning. The characteristic of moving frequently of vehicle nodes makes it very suitable for federated learning, and many researchers have applied federated learning to IoV. Lu et al. applied federated learning to vehicle network physical systems to protect data privacy [10]. With the emergence of federated learning, the privacy of users can be further protected while machine learning. Pokhrel and Choi proposed an autonomous blockchain-based federated learning (BFL) design for sensing privacy and efficient vehicular network [11]. Elbir and Sinem applied federated learning to vehicle networks to develop an intelligent transportation system [12]. Federated learning can

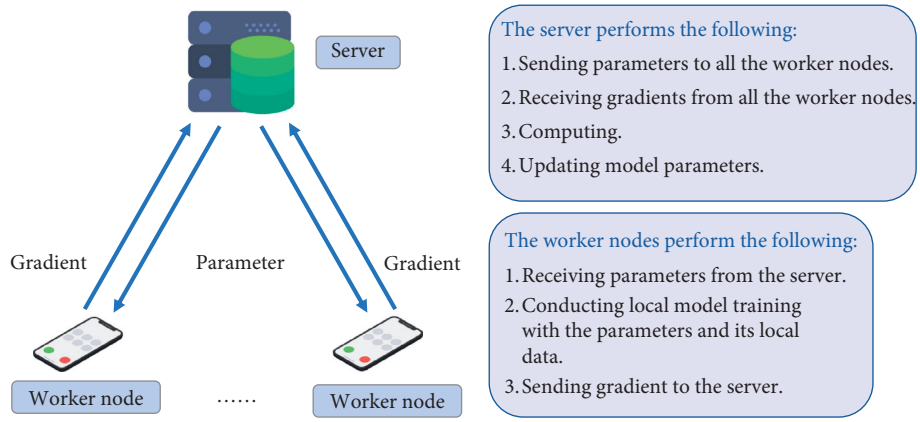


Figure 1: A traditional federated learning model.

make individual training data stay locally while building a model together, which greatly reduces the possibility of user's privacy leakage.

*2.2. Homomorphic Encryption.* Homomorphic encryption (HE) is a special encryption algorithm based on the computational complexity theory of mathematical problems, which allows the processing of encrypted data without the private key. In addition to the basic encryption operations, the biggest feature of homomorphic encryption is that it allows ciphertext to be calculated directly compared with the general encryption algorithm, and the results after decryption are the same as those directly calculated in the plaintext. Therefore, homomorphic encryption has a higher safety. As early as 1978, Ron Rivest and Leonard Adleman put forward the concept of homomorphic encryption. Homomorphic encryption can make the operation satisfy the additive homomorphism and multiplication homomorphism, and it is divided into somewhat homomorphic encryption and fully homomorphic encryption (FHE). The elliptic curvature cryptography (ECC) proposed by Neal Koblitz and Victor Miller in 1985 is a partial homomorphic encryption algorithm satisfying additive homomorphism. The famous RSA algorithm is also a partial homomorphic encryption algorithm satisfying multiplicative homomorphism. In 2009, Craig Gentry, a Ph.D. student from Stanford University, constructed a fully homomorphic encryption scheme based on the ideal lattice, which marks a breakthrough in homomorphic encryption technology.

Because of the characteristic of higher safety of homomorphic encryption, many scholars have applied it to the Internet of Things (IoT), biometric authentication, blockchain, and other fields for privacy protection. Zouari and Hamdi applied homomorphic encryption to IoT and realized the effective combination of information of multiple nodes safely [13]. Salem et al. applied homomorphic encryption to biometric identification, which improved the security of biometric authentication [14]. Besides, homomorphic encryption can also provide a method for privacy protection in blockchain, and She et al. integrate homomorphic encryption algorithm into blockchain and apply it to smart home systems to protect user's privacy [15]. Liang proposed a circuit copyright protection blockchain based on homomorphic encryption, which can effectively solve the security problem of circuit copyright transaction [16].

Homomorphic encryption provides a new method for privacy protection [17]. It can separate the ownership and processing rights of data and directly calculate the ciphertext, and the original data of any participant will not be leaked. This makes the user's privacy can be well protected.

*2.3. Blockchain.* Blockchain was originally revealed by researchers in 1991, aiming to add time stamps to digital documents so that they can be traced or cannot be tampered with. In 2009, Nakamoto applied blockchain to the management of the bitcoin financial system for the first time. The core of blockchain is distributed account, decentralized, smart contract, and consensus mechanism. Besides, the typical blockchain scheme also has the characteristics of a reliable database, detrust, trade quasi-anonymity, and open source programmable, so we can track data stored in blocks safely and transparently. In 2017, Wal-Mart, IBM, and JD.com launched a blockchain food safety project called Alliance in China. Maersk and IBM launched TradeLens to support information sharing and promote efficient and safe global trade. Also, blockchain can be used in electronic voting, copyright protection, and medical fields.

Blockchain has a higher safety, and it creatively uses hash calculation, proof of work, and distributed storage to make tamper with block almost impossible. At present, researchers have used blockchain to solve the problems of privacy protection in the Internet of Vehicles. In [18], Pokhrel and Choi applied blockchain and federated learning to autodriving to protect user's privacy and used a blockchain incentive mechanism to reward the worker nodes performed better in federated learning to encourage them to participate in federated learning more actively. Das et al. applied blockchain to vehicle theft prevention to ensure vehicle safety and owner's privacy through smart contracts [19]. In [20], Rawat applied blockchain to V2X communication to protect data privacy. Liu et al. proposed an electric vehicle power trading model based on blockchain and smart contracts [21].

The information records in blockchain have the characteristics of a lifelong responsibility system. Once

completed, it is very difficult to tamper with and delete. Because of the deterrent force of this lifelong responsibility system, commercial cooperation, social behaviour, and credibility will be greatly improved. It will be of great help to increase the construction of our future credit system and even the progress of human civilization. Finally, blockchain technology ensures that all parties to the collaboration see the same information system, which laid a good foundation for building a wider range of social cooperation in the future.

## 3. Privacy Protection Model and Algorithm

In the current application of IoV, V-C2X and MEC technology make the vehicles, people, and roadside units share information. It realizes the interaction of real-time traffic information which greatly facilitates our lives, but the privacy protection in IoV is not paid enough attention at present. The emergence of federated learning can well solve the problem in the application of the Internet of Vehicles at the present stage. It not only solves the problem of data silos but also ensures the security of user's privacy. To solve the privacy problems in IoV, we propose a blockchain-based federated learning pool framework [22–26].

As shown in Figure 2, our framework consists of the federated learning pool, adaptive learning models, and blockchain. We use blockchain as the bottom layer of the framework to ensure the reliability of data transmission. All parameters are uploaded to the server in the blockchain through base stations and RSU. Each adaptive learning model can automatically select the most suitable federated learning method according to the characteristics of the data source in the process of federated learning. And in V2V, vehicle nodes are equipped with CPC lightweight encryption algorithm to ensure the user's privacy and security while sharing information. Meanwhile, the use of homomorphic encryption algorithm in C2V is to further ensure security. In the end, the global model is built in the server of blockchain with multiple federated models.

*3.1. Federated Learning Pool.* Given the complexity of IoV, it is a very difficult problem to choose which federated learning method to use. However, if the server can independently analyse the distribution characteristics of data sources and automatically select the corresponding federated learning method, this problem will become very simple. Therefore, we propose the concept of federated learning pool, which is an adaptive federated learning module carried on servers and can select the most suitable federated learning method according to the characteristics of data sources. In FLP, according to practical application scenarios, some design criteria such as decentralizing or weak centralizing, energy-saving, and security are selected purposefully, and some strong or weak principles are selected, to improve the efficiency of the system [27, 28]. There are three federated learning methods (horizontal federated learning, vertical federated learning, and federated transfer learning) in our scheme, and the most suitable learning method will be chosen according to actual application scenarios, which protect the privacy of users.

*3.1.1. Horizontal Federated Learning.* In IoV, each vehicle is a collector of information, and all the real-time traffic information they collect is an indispensable part of building the global model. The information interaction between vehicles makes drivers get more information, but at the same time, user's data privacy may be leaked. In this model, we make vehicles as clients and servers in the blockchain as server nodes. The vehicles download the parameters from the blockchain and conduct local model training with the local data. After training, the updated parameters encrypted by CPC lightweight algorithm are returned to the blockchain. In this progress, the base station and RSU will aggregate the parameters returned by vehicles to share the work of the server.

The objective function of finite sum in this method is denoted by $\min_{w_{\text{vehicle}} \in \mathbb{R}^d} f(w_{\text{vehicle}})$, where $f(w_{\text{vehicle}}) \overset{\text{def}}{=} (1/n)\sum_{i=1}^{n} f_i(w_{\text{vehicle}})$ and $w_{\text{vehicle}}$ is the data parameter of vehicle.

We take $f_i(w_{\text{vehivle}}) = l(x_i, y_i; w_{\text{vehicle}})$; that is, the loss of prediction on example $(x_i, y_i)$ made with the data parameter of vehicle$w_{vehicle}$. We assume there are $K$ vehicle clients altogether, with $(DB_{\text{vehicle}})_k$ the set of local data on vehicle client $k$, with $n_k = |(DB_{\text{vehicle}})_k|$. Then, we can rewrite the objective function as

$$\begin{aligned} f(w_{\text{vehicle}}) &= \sum_{k=1}^{K} \frac{n_k}{n} F_k(w_{\text{vehicle}}), F_k(w_{\text{vehicle}}) \\ &= \frac{1}{n_k} \sum_{i \in DB_{\text{vehicle}}} f_i(w_{\text{vehicle}}). \end{aligned} \tag{1}$$

From the above definition, we can know that the total loss function is the weighted average of the local loss of each vehicle client and the number of samples. And in each round of communication, there will be a batch of gradient calculation, through multiple rounds of efficient iterative calculation to build a better model. The method of each iteration is as follows ($_t$ represents the $t$ th round of iteration).

Vehicle client performs as follows:

$$(\text{Avg}T_{\text{vehicle}})_k = \nabla F_k\left((w_{\text{vehicle}})^t\right), \tag{2}$$

that is, the average gradient for the local data with the current model parameters $(w_{\text{vehicle}})^t$.

The base station aggregates the gradient:

$$(w_{\text{vehicle}})^{t+1} \leftarrow (w_{\text{vehicle}})^t - \eta \sum_{k=1}^{K} \frac{n_k}{n} (\text{Avg}T_{\text{vehicle}})_k, \tag{3}$$

where $\sum_{k=1}^{K} (n_k/n)(\text{Avg}T_{\text{vehicle}})_k = \nabla f((w_{\text{vehicle}})^t)$ and $\eta$ represents the efficiency of machine learning.

Method of equivalent update:

$$\begin{aligned} &\forall k, (w_{\text{vehicle}})_k^{t+1} \leftarrow (w_{\text{vehicle}})^t - \eta(\text{Avg}T_{\text{vehicle}})_k, \\ &(w_{\text{vehicle}})^{t+1} \leftarrow \sum_{k=1}^{K} \frac{n_k}{n} (w_{\text{vehicle}})_k^{t+1}. \end{aligned} \tag{4}$$

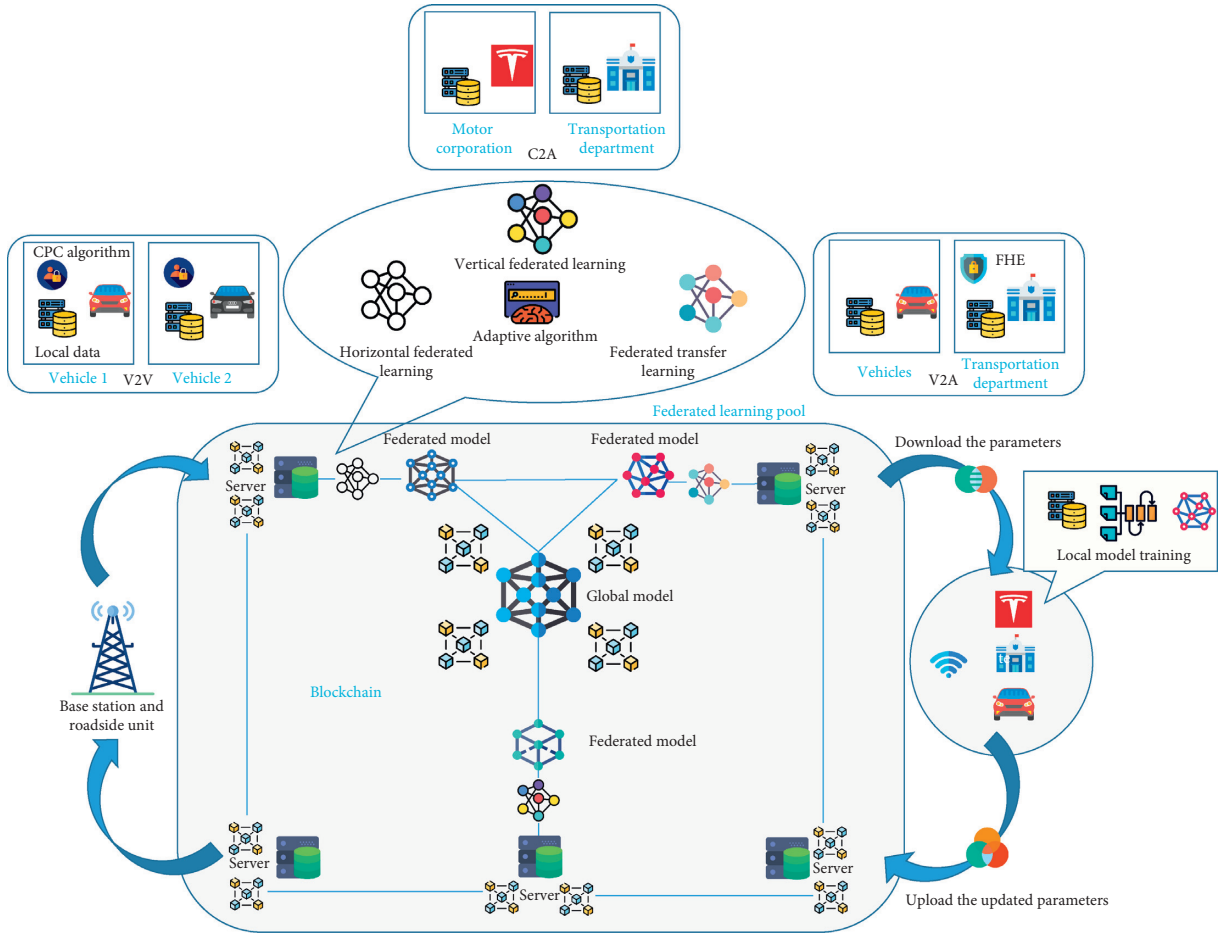


FIGURE 2: Blockchain-based federated learning pool.

The vehicle clients send parameters to the base station by several rounds of updating, and then, the base station computes weighted average of these parameters.

Algorithms 1 and 2 show the details of horizontal federated learning.

*3.1.2. Vertical Federated Learning.* The driver assistance system in IoV can provide a range of support for drivers while driving, including lane keeping assistance, autoparking assistance, brake assist, and automatic driving. These systems improve the driving experience and even avoid disasters in some dangerous situations. However, most manufacturers' driver assistance systems rely more on their sensors, cameras, and algorithms. Such a method can also train models, but the model lacks some other important data that is not perfect. Therefore, we apply vertical federated learning to some complex road conditions or accident-prone areas. The data collected by vehicle sensors in the same area are not the same as the data owned by the transportation department in characteristics and parameters. The vehicles and transportation departments use the data with different characteristics in the two data sets to train the local model and then send parameters to the base station and RSU server to conduct gradient polymerization. In the end, the updated parameters are sent to the server in the blockchain to build a federated model.

The data sets owned by vehicles and transportation department are different, and we assume the data owners were involved in building models as $P_t\,(t = 1, 2, \ldots, T)$. And suppose the data owners $P_t$ collaboratively train a machine learning model based on $D_j\,(j = 1, 2, \ldots, N)$ data samples $\{m_i, n_i\}_{i=1}^{N}$ and the feature vector $m_i \in \mathbb{R}^{Id}$ are distributed among $P_t$ parties, $\{m_i^t \in \mathbb{R}^{Id_t}\}_{t=1}^{T}$, where $d_t$ is the feature dimension of $P_t$. Without loss of generality, we assume $P_t$ holds the data labels. And we denote the data set of each part as $\mathfrak{F}_i^t \triangleq \{m_i^t\}$, for $t \in [T-1]$, $\mathfrak{F}_i^T \triangleq \{m_i^T, n_i^T\}$, and $\mathfrak{F}_i \triangleq \{\mathfrak{F}_i^t\}_{t=1}^{T}$, where $[T-1]$ denotes the set$\{1, \ldots, T-1\}$. Then, the vertical learning model can be described as

$$\min_{\Theta} \mathscr{L}(\Theta; \mathfrak{F}) \triangleq \frac{1}{N} \sum_{i=1}^{N} f(\rho_1, \ldots, \rho_T; \mathfrak{F}_i) + \lambda \sum_{t=1}^{T} \kappa(\rho_t), \quad (5)$$

where $\rho_t \in \mathbb{R}^{d_t}$ denotes the training parameters of the $t$ th parties, $\Theta = [\rho_1, \rho_2, \ldots, \rho_T]$, $f(\cdot)$ and $\kappa(\cdot)$ denotes the loss function and regularizer, and $\lambda$ is the threshold. In this condition, the loss function can be described as follows:

$$f(\rho_1, \ldots, \rho_T, \mathfrak{F}_i) = f\left(\sum_{t=1}^{T} m_i^t \rho_t, n_i^T\right). \quad (6)$$

**Input**: update parameters encrypted by CPC $(w'_{\text{vehicle}})_k$
**Output**: update parameters of the vehicle $U_{\text{vehicle}}$
1: Decrypt $(w'_{\text{vehicle}})_k$;
2: Initialize $(w_{\text{vehicle}})_0$;
3: **for** each round $t = 1, 2, \ldots$ **do**
4: $e \longleftarrow \max((\text{Percent}_{\text{vehicle}}) \cdot (K), 1)$;
5: //$\text{Percent}_{\text{vehicle}}$ is percentage of vehicles selected each round.
6: $C_t \longleftarrow$ (random set of $e$ vehicles);
7: **for** each vehicle $k \in C_t$ in **parallel do**
8: $(w_{\text{vehicle}})_k^{t+1} \longleftarrow \text{VehicleUpdate}(k, (w_{\text{vehicle}})^t)$;
9: **end for**
10: $(w_{\text{vehicle}})^{t+1} \longleftarrow \sum_{k=1}^{K} / n_k n (w_{\text{vehicle}})_k^{t+1}$;
11 **end for**

Algorithm 1: Horizontal federated learning: the base station executes.

**Input:** $DB_{\text{vehicle}}$
**Output:** $w_{\text{vehicle}}$
1: $g \longleftarrow$ (split $DB_{\text{vehicle}}$ into batches of size $G$); //$G$ is local minibatch size of vehicle.
2: **for** each local epoch $i$ from 1 to $I_{\text{vehicle}}$ is the number of local epochs.
3: **for** batch $b \in \mathcal{G}$ **do**
4: $w_{\text{vehicle}} \longleftarrow w_{\text{vehicle}} - \eta \nabla l\ (w_{\text{vehicle}};\ b)$;
5: **end for**
6: Use CPC to encrypt $w_{\text{vehicle}}$;
7: return $w'_{\text{vehicle}}$ to the base station
8: **end for**

Algorithm 2: Horizontal federated learning: VehicleUpdate $(k, w_{\text{vehicle}})$.

The objective is for $P_t$ to find its $\rho_t$ without sharing its data set $\mathfrak{F}_i^t$ or parameter $\rho_t$ to other parties.

Then, we describe the way the server updates parameters. If a minibatch $L \subset \mathfrak{F}$ of data is sampled, the stochastic gradient $\rho_t$ is given by

$$\varpi_t(\Theta; L) \triangleq \nabla_t f(\Theta; L) + \lambda \nabla \kappa(\rho_t). \tag{7}$$

For the arbitrary loss function, let $H_i^t = m_i^t \rho_t$, $H_i = \sum_{t=1}^{T} H_i^t$, and the collection of information needed to calculate the loss function $\nabla_t f(\Theta; L)$ is defined as

$$H_{-t}^L \coloneqq \left\{H_p^t\left(\rho_p, L^p\right)\right\}_{p \neq t}, \tag{8}$$

where $H_p^t(\cdot)$ is a function summarizing the information required form data owner $p$ to $t$.

Based on the descriptions above, the stochastic gradients can be computed by (7) as

$$\varpi_t(\Theta; L) = \nabla_t f\left(H_{-t}^L, \rho_t; L\right) + \lambda \nabla \kappa(\rho_t) \triangleq \varpi_t\left(H_{-t}, \rho_t; L\right). \tag{9}$$

In each iteration of the server, the following formula is used to update the characteristic dimension of the $t$ th data party ($\eta$is the learning rate):

$$d_t = -\eta \varpi_t\left(H_{-t}, \rho_t; L\right). \tag{10}$$

Because $H_{-t}$ is the intermediate information obtained in the most recent synchronization, which may contain staled information so it may no longer be an unbiased estimate of the true partial gradient. For another thing, during the $Q$ local updates, no interparty communication is required. In the same spirit, a sequential version of the algorithm allows two parties to update their local $\rho_t$ sequentially, while each update consists of $Q$ local updates without interparty communication.

Algorithm 3 shows the details of vertical federated learning.

*3.1.3. Federated Transfer Learning.* Federated transfer learning is suitable for learning different data sets from different feature spaces. It migrates the features of different feature spaces to the same potential representation and train models with the labels in the labelled data collected by different parties. The goal is to use federated transfer learning to solve the problem of the lack of data and labels while protecting privacy. It can not only be applied to two sample spaces but also two different data sets. The federated transfer learning selects overlapping data and then conducts sample alignment, which is helpful for the part with good labelled data to build an improved model. Based on this model, the other parties predict the lack of characteristics of the samples and modify the model.

In IoV, the distribution of the data collected by vehicles and the data owned by the transportation department well meets the above conditions. The data collected by vehicles

**Input:** $\eta$
**Output:** $\rho_1, \rho_2, \ldots, \rho_T$
1: Party $P^t(t = 1, 2, \ldots, T)$ initialize $\rho_1, \rho_2 \ldots \rho_T$;
2: **Exchange** $P^t(t = 1, 2, \ldots, T)$;
3: **for** *each iteration* $x = 1, 2, \ldots$ **do**
4: Randomly sample a minibatch $L \subset \mathfrak{J}$
5: **for** *each party* $t = 1, 2, \ldots, T$ *sequentially* **do**
6: **for** *each local iteration* $r = 1, 2, \ldots, Q$ **do**
7: $t$ computes $\varpi_t(H_{-t}, \rho_t; L)$ using (9);
8: update $\rho_t \leftarrow \rho_t - \eta \varpi_t(H_{-t}, \rho_t; L)$;
9: **end for**
10: **Exchange** $P^t$ $(t = 1, 2, \ldots, T)$;
11: **end for**
12: **end for**

Algorithm 3: Vertical federated learning.

are often road conditions in a certain region, and the data owned by the transportation department contain other data except that. If the leakage of data occurs in the process of interaction between the two parties of the training model, it will cause very serious problems. According to the distribution characteristics of vehicle data sets and the transportation department, we choose federated transfer learning to build a model and protect privacy.

Let us denote data source of vehicles $V$, data source of transportation $T$, servers of RSU Server-RSU, servers in the blockchain Server-BC. Firstly, the hidden neural networks of $V$ and $T$ are established by an end-to-end solution. We initialize the parameters to be passed in $V$ and $T$ and use a prediction function to mark the target domain and minimize the alignment loss between $V$ and T. Thus, we can express the function of updating of Server-RSU. We can also calculate the update gradient passed to $V$ and $T$ in the diffusion of Server-RSU to update and optimize the existing data model of $V$ and $T$. To meet the security and privacy requirements of federated transfer learning, it is required to ensure that original data of $V$ and $T$ are hidden. We use the fully homomorphic encryption method to protect the privacy. $V$ and $T$ will generate their public keys and encrypt the parameters needed to be passed in each data party. At the same time, the private key is used to decrypt locally to obtain training information which will be uploaded to Server-BC. The notations used in Algorithm 4 are listed in Table 1.

*3.2. Federated Learning Model.* In this section, we will describe the details of our lightweight encryption algorithm and the adaptive learning models.

*3.2.1. CPC Lightweight Encryption Algorithm.* Although federated learning has brought many public values (such as protecting privacy and breaking data silos), it still has some malpractice. In federated learning, although the user's data do not leave the device and only parameters of model and gradients returned by users are transmitted in the channel, these gradients almost carry all the information of the user's data, and we can infer the user's information through reverse engineering and other methods. In federated learning, both server and worker can repeatedly get the model parameters after each iteration, which means that it is easier to infer the user's data in federated learning. Therefore, based on the Feistel cipher structure [29], we propose a new lightweight encryption algorithm called CPC. The CPC lightweight encryption algorithm has fewer encryption rounds, simpler conversion, and more efficient replacement. Considering the poor performance of the vehicle system, to reduce the cost of encryption of vehicle nodes, the algorithm only performs eight rounds of encryption, in which the input is 256-bit plaintext and 256-bit master key. Figure 3 shows the process of our encryption algorithm. We divide the plaintext into four subblocks and use the master key to derive a subkey, and then, four new subblocks are output in each round of conversion.

*(1) The Generation of the Master Key.* To encrypt the message, each sensor node generates an encryption key called $K_{\text{enc}}$, which is shared it with the base station. In this section, we present the process of the generation of this key which will be used in the CPC encryption process. Table 2 shows the parameters used in the key generation process.

Before deployment, the base station predistributes a set of keys to each sensor node in the network. Our asymmetric keys are based on elliptic curve cryptography (ECC) algorithm. This cryptography has already been proved that it provides a more security of shared keys, which ensures an equivalent level, or even more secure, than other asymmetric systems. The base station begins to create an asymmetric key pair unique to each node in the network. For example, a node $N$ will have a pair of unique public-private keys $(P_n, K_n)$. $K_n$ is selected in the interval $[1, m]$, where $m$ is the parameters of ECC. $K_n$ is considered as the private key of node $N$. The public key of $N$ is obtained by the following scalar multiplication:

$$P_n(x_N, y_n) = K_n \cdot G(x, y), \tag{11}$$

where $G(x, y)$ is the point on the curve.

Each sensor node stores its public and private keys, identity ID, and public keys $P_{\text{BS}}$ of the base station:

**Input:** $S_v$, $S_T$
**Output:** $l_{Server-RSU}$
1: **V and T do**
2: Initialize $\mathcal{N}$ required by $\mathcal{L}$;
3: Distribute the public keys $\tau_V$ and $\tau_T$;
4: Use fully homomorphic encryption to encrypt $\mathcal{N}$;
5: Send $[\mathcal{N}]$ to Server-RSU;
6: **Server-RSU do:**
7: Distribute the public keys $\tau_{RSU}$;
8: **while** the model is not convergent **do**
9: **Server-RSU Execute:**
10: Decrypt $[\mathcal{N}']$locally;
11: Calculate $\mathcal{L}$, $\partial_v$, $\partial_T$;
12: Use fully homomorphic encryption to encrypt $\partial_v$, $\partial_T$;
13: Send $[\partial_v]$ to V and send $[\partial_T]$ to $T$;
14: Send $[\mathcal{L}]$ to Server-BC;
15: $V$ **do**:
16: Decrypt $[\partial_V]$ locally;
17: Calculate $\gamma_V$;
18: Use fully homomorphic encryption to encrypt $\gamma_V$;
19: Send $[\gamma_V]$ to Server-RSU;
20: $T$ **do:**
21: Decrypt $[\partial_T]$ locally;
22: Calculate $\gamma_T$;
23: Use fully homomorphic encryption to encrypt $\gamma_T$;
24: Send $[\gamma_T]$ to Server-RSU;
25: **end while**

Algorithm 4: Federated transfer learning.

Table 1: Notations and definitions.

| Notation | Definition |
|---|---|
| $\mathcal{S}_V$ | Hidden neural network of vehicles |
| $\mathcal{S}_T$ | Hidden neural network of transportation department |
| $\ell_{\text{Server-RSU}}$ | Updated parameters of RSU server |
| $\mathcal{N}$ | The initialization set of parameters in two neural networks needed by RSU server |
| $\mathcal{N}'$ | Set of parameters in two neural networks needed by terrestrial node servers |
| $\mathcal{L}$ | Function of updating in RSU server |
| $\tau_V$ | Public key of hidden neural network for the data sets of vehicles |
| $\tau_T$ | Public key of hidden neural network for the data sets of transportation |
| $\tau_{\text{RSU}}$ | Public key of RSU server |
| $\partial_V$ | Updated parameter returned to vehicles by RSU server |
| $\partial_T$ | Updated parameter returned to transportation department by RSU server |
| $[\cdot]$ | Fully homomorphic encryption function |
| $\gamma_V$ | Set of parameters that the hidden neural network of vehicles sent to RSU server |
| $\gamma_T$ | Set of parameters that the hidden neural network of transportation department sent to RSU server |

$(P_n, K_n, I\ D, P_{\text{BS}})$. The base station stores its public and private keys $(P_{\text{BS}}, K_{\text{BS}})$ and all the public keys $P_{\text{N}}$ of different sensors in the network.

After deploying sensors, each node $N$ must create a master encryption key $K_{\text{enc}}$ to encrypt messages in the encryption process. Our main encryption key $K_{\text{enc}}$ consists of 256 bits, which contains two components (Figure 4). The first component $K_{\text{ECC}}$ is composed of the first component (128 bits) of the output of the ECC cipher algorithm. The second component is a 64-bit ID (node identifier). When these components are generated, $K_{\text{ECC}}$ are combined with ID to generate a 256-bit master encryption key $K_{\text{enc}}$ through SM3.

After the initial deployment is completed, the identity authentication and key distribution stage are carried out. We intend to use the preassigned asymmetric key before the deployment of nodes to share the key between each sensor node and the base station. In the process of key generation, we use Diffie–Hellman mechanism [30]. The main idea is that each node in network can create a shared key $K_{\text{ECC}}$ by using the parameters of ECC without interacting with the

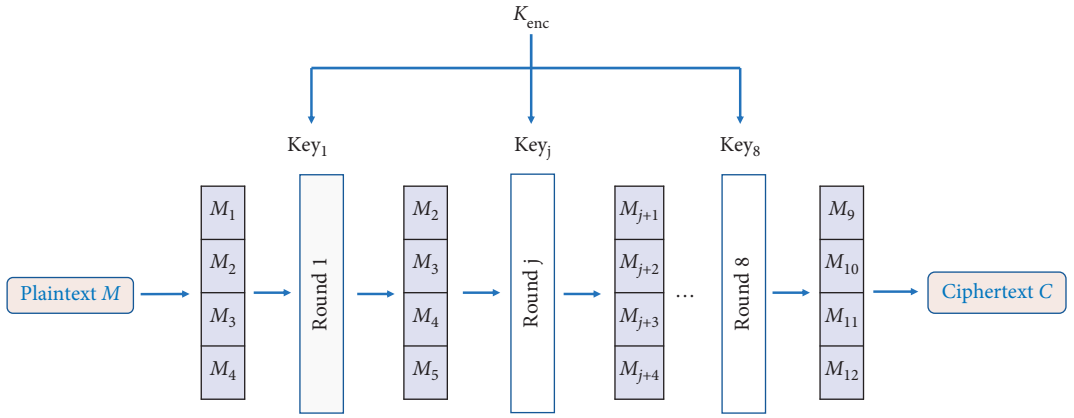


Figure 3: Progress of CPC encryption algorithm.

Table 2: Key generation parameters.

| Parameter | Description |
|---|---|
| $G$ | Base point that lies on the elliptic curve |
| $K_N$ | Private key of sensor node $N$ |
| $K_{enc}$ | The main encryption key with 256 bits |
| $K_{ECC}$ | Key produced with the ECC algorithm (the first component of $K_{enc}$ with 128 bits) |
| $H_{SM3}$ | SM3 hashing function (output of 256 bits) used to hash the concatenation result of $K_{ECC}$ and $ID_N$ |
| $t_N$ | A nonce generated by sensor node $N$ |
| $MAC_K(M)$ | Message authentication code (MAC) of message $M$ using MAC key $K$ |
| $N \longrightarrow$ BS: $M$ | Node $N$ sends a message $M$ to BS |

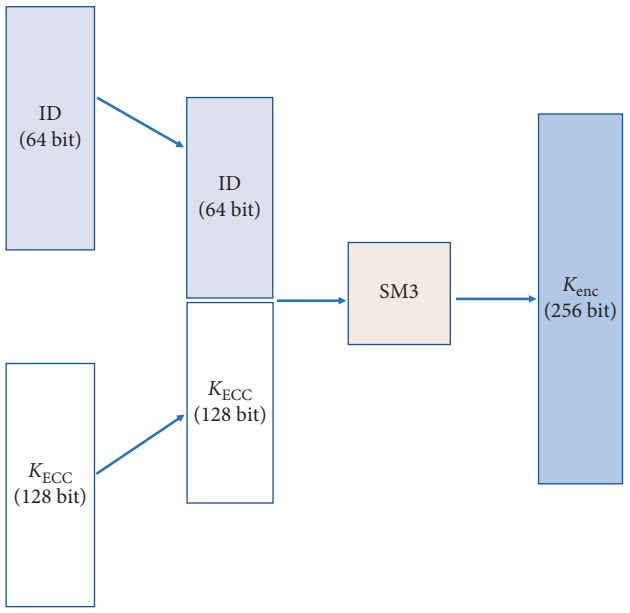


Figure 4: Generation of master encryption key $K_{enc}$.

base station, which means no message there is exchanged. The generation process of $K_{ECC}$ is as follows.

The base station (BS) calculates a temporary key $D_{BS-N}$:

$$D_{BS-N} = K_{BS} \times P_N. \tag{12}$$

Node $N$ calculates a temporary key:

$$D_{N-BS} = K_N \times P_{BS}. \tag{13}$$

According to the Diffie–Hellman mechanism,

$$\begin{aligned} K_{ECC} = D_{N-BS} &= K_N \times P_{BS} \\ &= K_N \times (K_{BS} \times G) \\ &= (K_N \times G) \times K_{BS} \\ &= P_N \times K_{BS} \\ &= D_{BS-N}. \end{aligned} \tag{14}$$

Combine $K_{ECC}$ and ID and then hash the merged result by using SM3 (a hash function) to generate our encryption key $K_{enc}$ (output 256 bits):

$$K_{enc} = H_{SM3}(IDK_{ECC}). \tag{15}$$

Finally, the node $_N$ sends its main encryption key $K_{enc}$ to the base station safely with the message containing authentication:

$$N \longrightarrow \text{BS: } ID_N \| ID_{BS} \| MAC_{K_N}(K_{enc}, ID_N \| ID_{BS} \| t_N). \tag{16}$$

After these processes, $K_{enc}$ will be stored in the memory of the sensor node. With the length of 256 bits, an attacker cannot figure out $K_{enc}$ and this key will keep secret during the encryption process.

1: Split the 256-bit main encryption key $K_{enc}$ into 8 blocks of 32-bits: $K_1$, $K_2$,. . ., $K_8$
2: The 8 subkeys ($Key_1$, $Key_2$, . . ., $Key_8$) are computed as follows:
3: $Key_1 := K_1 \oplus [Inv(K_2) \oplus Inv(K_3)]$
4: $Key_2 := Key_1 \oplus [Inv(K_3) \oplus Inv(K_4)]$
5: $Key_3 := Key_2 \oplus [Inv(K_4) \oplus Inv(K_5)]$
6: $Key_4 := Key_3 \oplus Key_2 \oplus Key_1$
7: $Key_5 := Key_4 \oplus [Inv(K_5) \oplus Inv(K_6)]$
8: $Key_6 := Key_5 \oplus [Inv(K_6) \oplus Inv(K_7)]$
9: $Key_7 := Key_6 \oplus [Inv(K_7) \oplus Inv(K_8)]$
10: $Key_8 := Key_7 \oplus Key_6 \oplus Key_5$

Algorithm 5: Subkeys generation algorithm.

*(2) Generation of the Subkey*. We use the master key $K_{enc}$ to calculate eight subkeys $Key_i$, which is used for the $i$ round encryption. In the generation process of subkey, we design some calculations to ensure that different subkeys have different characteristics when facing a particular attack. In the initialization phase, we split the 256-bit master key $K_{enc}$ into 8 equal-length 32-bit blocks $K_1, K_2, \ldots, K_8$. The generation of each subkey is based on different master key blocks $K_i$ and different exclusive-OR functions, and the function Inv $(k)$ is used to reverse the different positions of $k$. The generation details of different subkeys are given in Algorithm 5.

*(3) Progress of Encryption*. The content of each round in the process of encryption is shown in Figure 5. Each round contains four different operations: Count_Zero, function $f_1$, function $f_2$, and permutation operation. 128-bit plaintext $M$ is divided into four 32-bit blocks. Four new bit sequences are generated in each round and then begin the next round of encryption.

Algorithm 6 shows the details of encryption.

In the end, three blocks $M_{j+1}, M_{j+2}, M_{j+3}$ are replaced to update the order of the encrypted text blocks. Algorithm 7 shows the permutation function in the process of encryption.

At the end of the eighth round of encryption, the generated ciphertext $C$ is defined as follows:

$$C = M_9 \| M_{10} \| M_{11} \| M_{12}. \tag{17}$$

*(4) Progress of Decryption*. When the base station receives the ciphertext with the authentication message sent by the nodes, separates it, and then starts the process of decryption, it will use the shared key $K_{enc}$ to decrypt the ciphertext to obtain the plaintext. The process of decryption is shown in Figure 6.

In the Feistel cryptosystem, decryption is an inverse process of encryption. It uses the same functions and parameters in the process of encryption. There is no essential difference between encryption and decryption calculations, but the order of using the subkey sequence $j$ is on the contrary. We assume that the sequence of blocks in the

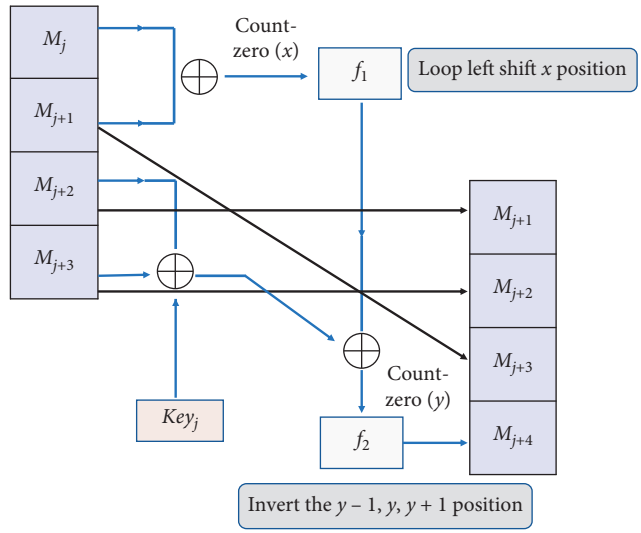


Figure 5: Progress of encryption.

process of encryption is known, and Figure 7 shows the main steps of each round in the decryption process.

The permutation function in the process of decryption is shown in Algorithm 8.

The definition of plaintext is as follows:

$$M = C_1 \| C_2 \| C_3 \| C_4. \tag{18}$$

The process of decryption is shown in Algorithm 9.

*3.2.2. Adaptive Model.* In our federated learning framework, there are three adaptive learning models, V2V, V2A, and C2A, according to the characteristics and scenarios of data sources. And servers in the 3 models are all the servers in the blockchain. In our models, adaptation is mainly reflected in two aspects. On the one hand, the system allows different federated learning methods to meet the application requirements of different scenarios. Users can assign the initial learning mechanism through some APIs. On the other hand, the system supports the upgrading of federated learning methods at any time to achieve a hot upgrading of federated learning mechanism:

**Input:** $M$, $K_{enc}$
**Output:** $C$
1: Divide $M \longrightarrow M_1, M_2, M_3, M_4$
2: Generate different $Key_j$ form the main Key $K_{enc}$
3: **for** each round $j = 1:8$ **do**
4: $X := M_j \oplus M_{j+1}$
5: $Y := M_{j+1} \oplus M_{j+3}$
6: $Count_1 := Count - Zero\ (X)$
7: $W := f_1\ (Count_1, X)$
8: $R := W \oplus Y$
9: $Count_2 := Count - Zero\ (R)$
10: $S := f_2(Count_2, R)$
11: $M_{j+4} := S$
12: Permutation: $(M_{j+1}, M_{j+2}, M_{j+3})$
13: **end for**
14: $C := M_9||M_{10}||M_{11}||M_{12}$
15: Output C

Algorithm 6: CPC encryption process.

1: Let $t_r$ a temporary variable
2: $t_r := M_{j+2}$
3: The permutation operations are the following:
4: $M_{j+2} := M_{j+3}$
5: $M_{j+3} := M_{j+1}$
6: $M_{j+1} := t_r$

Algorithm 7: Permutation function for round $j$ in the encryption process.

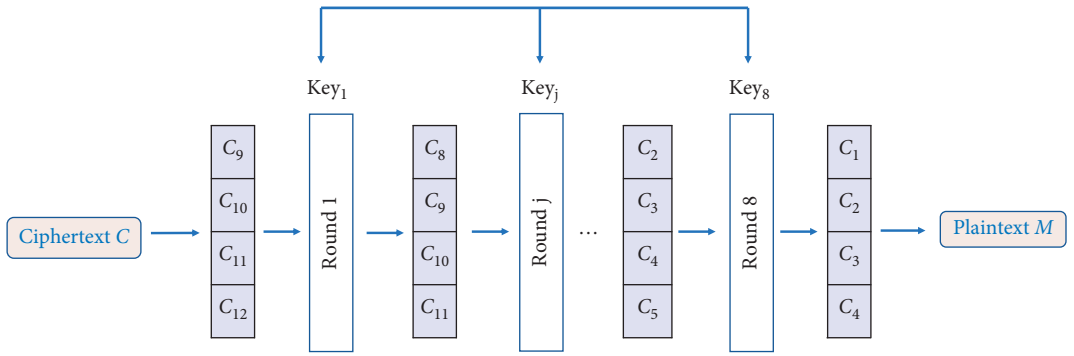


Figure 6: The overall architecture of decryption.

(1) V2V

In this model, we make vehicles as the worker nodes. At the same time, we apply the CPC lightweight encryption algorithm to the vehicle system.

Step 1: vehicles download the parameters from the blockchain and conduct local data training.
Step 2: when the training is completed, the gradient is encrypted through the CPC encryption algorithm and returned to the base station and RSU servers.
Step 3: the base station and RSU servers conduct gradient aggregation and upload the updated parameters.
Step 4: the servers in blockchain choose horizontal federated learning to build a model through the FLP module according to the distribution characteristics of data sources.

(2) C2A

In this model, we make the motor corporation and transportation department as the worker nodes.

Step 1: motor corporation and transportation department download the parameters from the blockchain and conduct local data training.
Step 2: when the training is completed, the gradient is returned to the base station and RSU servers.

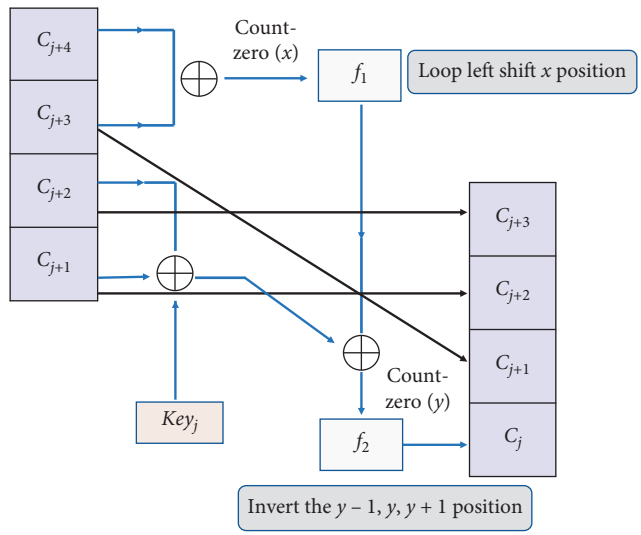


Figure 7: The progress of decryption.

1: Let $d_r$ a temporary variable
2: $d_r := C_{j+1}$
3: The permutation operations are the following:
4: $C_{j+1} := C_{j+3}$
5: $C_{j+3} := C_{j+2}$
6: $C_{j+2} := d_r$

Algorithm 8: Permutation function for round $j$ in the decryption process.

**Input:** $C$, $K_{enc}$
**Output:** $M$
1: Receiving C
2: Generate different the 8 subkey $Key_j$ from the main Key $K_{enc}$
3: Partition C into 4 segments: $C_9$, $C_{10}$, $C_{11}$, $C_{12}$
4: **for** each round $j = 8:1$ **do**
5: Permutation: $(C_{j+!}, C_{j+2}, C_{j+3})$
6: $X := C_{j+4} \oplus C_{j+1}$
7: $Y := M_{j+2} \oplus M_{j+3} \oplus Key_j$
8: $Count_1 := Count - Zero\ (X)$
9: $W := f_1\ (Count_1, X)$
10: $R: W \oplus Y$
11: $Count_2 := Count - Zero\ (R)$
12: $S := f_2\ (Count_2, R)$
13: $C_j := S$
14: $i := i - 1$
15: **end for**
16: $M := C_1||C_2||C_3||C_4$
17: Output $M$

Algorithm 9: CPC decryption process.

Step 3: the base station and RSU servers conduct gradient aggregation and upload the updated parameters.
Step 4: the servers in blockchain choose vertical federated learning to build a model through the FLP module according to the distribution characteristics of data sources.

(3) V2A

In this model, we make vehicles and the transportation department as the worker nodes. At the same time, we apply the fully homomorphic encryption algorithm to protect privacy while the transportation department is sharing the data.

Step 1: vehicles and transportation department download the parameters from the blockchain and conduct local data training.
Step 2: when the training is completed, the gradient of the transportation department is encrypted by FHE and returned to the base station and RSU servers.
Step 3: the base station and RSU servers conduct gradient of the transportation department and vehicles aggregation and upload the updated parameters.
Step 4: the servers in blockchain choose federated transfer learning to build a model through the FLP module according to the distribution characteristics of data sources.

In the end, the global model is built in the blockchain with multiple federated models.

## 4. Experiment and Analysis

In this section, we will discuss the security of our model and analyse the results of simulated experiments. We tested and compared the performance efficiency and storage cost of the CPC lightweight encryption algorithm with other encryption algorithms and analysed the advantages of the FLP module compared to traditional distributed machine learning based on the logistic regression model in the simulated environment of IoV. We also used caliper to test the performance of our system. In the end, we analysed the security and privacy of our proposed framework from the perspective of potential attacks. The details are as follows.

In our experiments, we used the TOSSIM simulator which is used for wireless sensor networks (WSNs) [31] to build the vehicle sensor-base station model. To measure the execution time of CPC and other encryption algorithms, we tested a variety of encryption algorithms in the TinyOS [17] system through many experiments, including CPC, TEA [32], XTEA [33], and AES [34], which are all newly proposed algorithms and used for low-resource embedded devices or mobile devices. Novelan et al. gave the implementation of the encryption algorithm TEA. TEA adopts a simplified scrambling function and short data packet, which can greatly reduce the cost of the decryption and encryption. However, it presents a poor key agility (the amount of time from generating/importing a new key to starting encrypting is negligible) and increases the burden of the gateway node. XTEA is the latest variation in TEA, which is an encryption algorithm based on ECC with a 128-bit key, 64-bit block, and 64 rounds Feistel structure. With the help of ECC, XTEA greatly improves the lifetime network, but the memory size remains a big challenge for this algorithm. Compared with the former algorithms, AES reduces the storage of keys in node memory by using master keys in key establishment. But resistance to attack is low, since the master key can be compromised at any time, and the different keys established after deployment using this key can be compromised too.

It is extremely important for critical applications in sensor networks to have a security mechanism that ensures authentication and confidentiality. However, optimal security in this type of network is particularly difficult due to the limitation of node resources. Therefore, we compared it with the above algorithms by experiments for testing our algorithm. To make the comparison, two crucial parameters have been selected: performance efficiency and storage cost. Then, we used the spark analysis platform to simulate the IoV environment and made analysis and prediction of the data. The specific experimental environment is shown in Table 3.

Firstly, we measured the performance efficiency of CPC algorithm. Because the decryption time is the same as the encryption time, we only measured the execution time of encryption. We used the above encryption algorithms to encrypt 50 times each and recorded the execution time through different lengths of vehicle sensor data plaintext. Figure 8 shows the time of the four algorithms to encrypt different plaintexts.

As shown in Figure 8, the encryption time the CPC algorithm needs is shorter compared with other tested algorithms. We can conclude that the execution time provided by CPC is much faster than other algorithms. This is due to the use of simple mathematical XOR, scrambling functions, and a less encryption round of CPC to be completed.

Then, we measured the storage cost of CPC algorithm. Figure 9 presents the result of the comparison in terms of occupying memory space by the four algorithms. During simulation, only the encryption times were measured.

As shown in Figure 9, TEA algorithm uses the longest S-box and can encrypt data with maximum length of 64 bits, which is present more times than CPC. XTEA algorithm takes more times than CPC to execute the encryption process because it contains more rounds. So CPC has superiority over the three algorithms.

The information obtained by sensors has certain limitations because the actual driving of the vehicle is restricted to time and space. The application of federated learning effectively improves the robustness of the machine learning model in IoV. In the following experiments, we compared the model and algorithm proposed in this paper with the traditional distributed machine learning algorithm based on the logistic regression model. Based on the analysis of multiple orders of magnitude of IoV data sets, we predicted the road conditions of next week. Meanwhile, we tested the success rate of avoiding obstacles. The test results are shown in Figures 10 and 11.

Table 3: The experimental environment.

| Number of cluster servers | 5 |
|---|---|
| CPU | AMD Ryzen 5900x |
| GPU | NVIDIA GeForce GTX 2080Ti |
| Memory of single pc | 16G |
| System version | Ubuntu 18.04 |
| Spark version | 2.4.2 |
| Blockchain architecture | Fisco-Bcos |
| Federated learning framework | FATE |
| API | Pyspark |

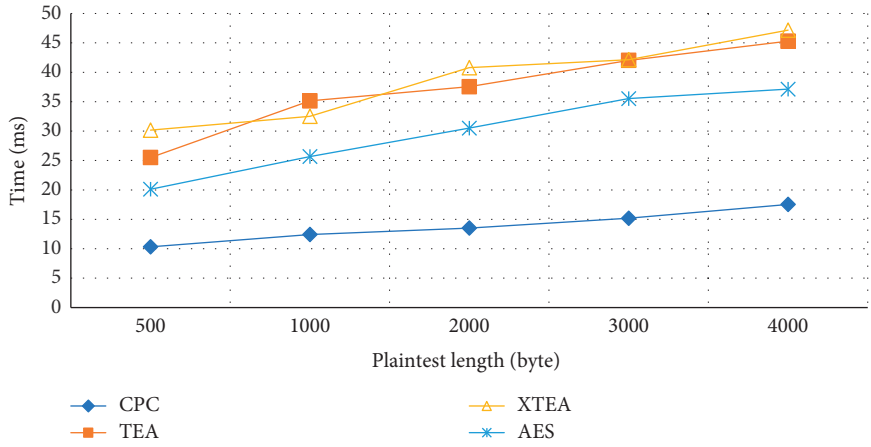


Figure 8: Performance efficiency comparison.

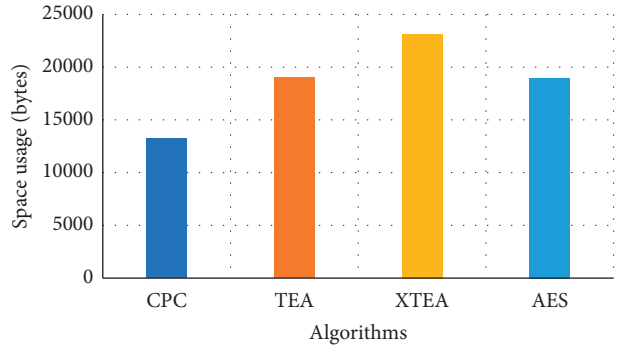


Figure 9: Storage cost comparison.

As shown in Figures 10 and 11, in the test of predicting road conditions and avoiding obstacles, with the increase in the magnitude of the input data set, the success rates of the two algorithms are also improved. And with the same data scale, the success rates of two tests based on FLP are higher than the traditional distributed machine learning algorithm based on the logistic regression model, which means our model is better in terms of accuracy.

Then, we measured the average computation time of FLP and the traditional distributed machine learning model in two experiments mentioned. Figure 12 shows the results.

As shown in Figure 12, with the same data scale, FLP takes about 15% less time than the traditional distributed machine learning algorithm based on the logistic regression model, which means our model is better in terms of effectiveness.

Due to the dependence and mobility on massive data, the performance index of blockchain-driven IoV network is quite important, which includes latency, energy consumption, throughput, and scalability. In our experiment, we used the caliper to test the performance. Caliper is a blockchain performance testing framework that currently supports testing for processing traffic (TPS), latency, and resource utilization. After each round of test, users can obtain a series of test results and reports by caliper.

As shown in Figure 13, the throughout increased steadily with the increase in transaction times. It reached the peak when the transaction times reached 5000, the throughput is

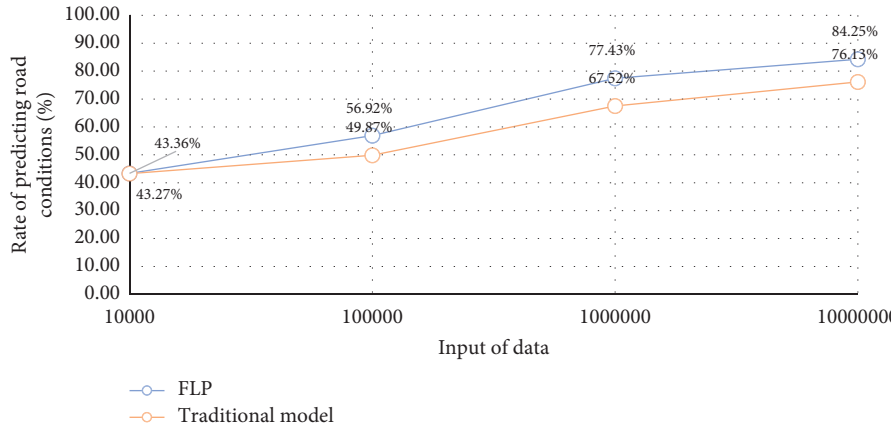


Figure 10: Accuracy rate of traffic prediction.

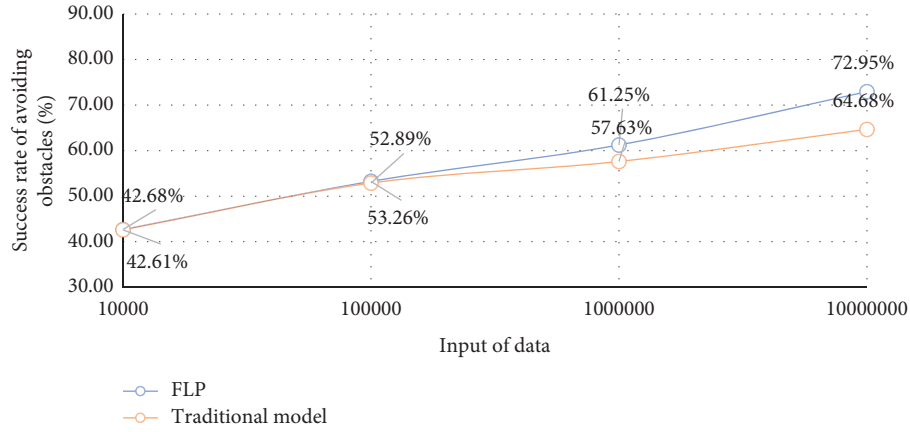


Figure 11: Success rate of obstacle avoidance test.

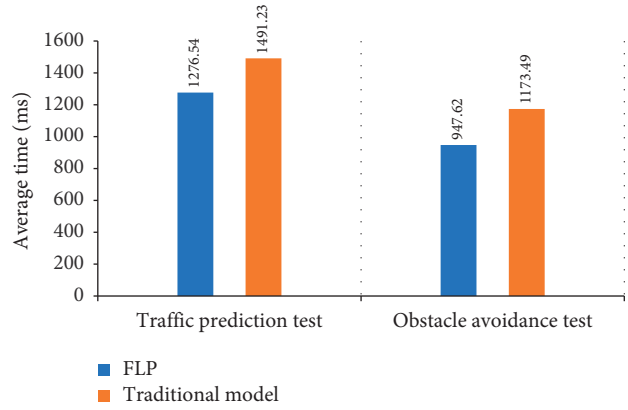


Figure 12: Performance efficiency comparison.

296.4 TPS, and the average latency is 215.4 ms. Then, it began to decline slowly when the transactions times exceed 5000. At present, there is no national standard for blockchain performance indicators, and China Institute of Information and Communications are actively formulating it. According to the existing blockchain industry standards (Table 4), the performance of our system meets the requirements.

In the edge network, on the one hand, due to the limitation of calculation, bandwidth resources, and the distributed characteristics of network structure, it has always

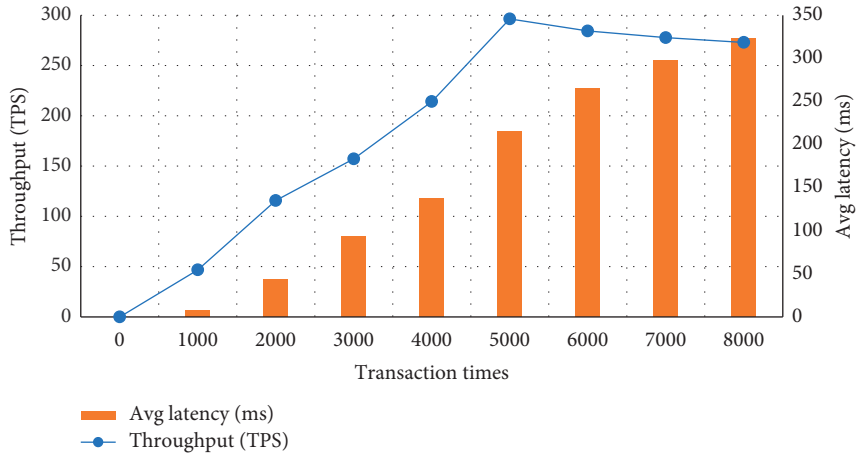


Figure 13: Performance evaluations.

Table 4: Blockchain industry standards.

| Name | Requirement |
|---|---|
| Success rate | >95% |
| Average response time | <0.5 s |
| Average latency | <1 s |
| Throughput (TPS) | 200~300 |
| Success rate | >95% |

been the focus of current research to effectively mine and utilize the distributed data of multisource and heterogeneous in the network. On the other hand, data sharing faces serious privacy disclosure risk. Once the data provider shares the data, it will lose the control of the data and face higher security risks. The application of licensed blockchain establishes a secure cooperation mechanism among distrusted parties. By embedding FLP into the consensus protocol process of licensed blockchain, some security risks can be alleviated:

(i) Remove centralized trust entities: the traditional access control model based on RBAC [35] and ABAC [36] relies on a central server to complete the management and judgment of permissions, which is easy to be attacked by illegal users to make data and files leaked. Licensed blockchain replaces trusted centralized management servers and connects each participant through multiparty data retrieval. In the proposed blockchain-driven data sharing scheme, centralized trust entities are no longer needed. Our model adopts an access control policy based on smart contracts to ensure access effectiveness. The smart contracts are stored in the blockchain and cannot be changed once published, and they will start once the conditions are met. Therefore, we can ensure the transparency and nontampering of the access control policy by smart contracts, which reduces the risk of data leakage caused by centralized trust.

(ii) Ensure the quality of shared data: in order to prevent dishonest data providers from sharing invalid data, the consensus mechanism based on federated learning pool will verify the quality of data models learned by other data providers, and only qualified data sets and models will be retained.

(iii) Safe data management: only data retrieval information will be uploaded to the licensed blockchain, while real data are stored locally. Data owners can control their data permissions by changing retrieval information. Meanwhile, in our framework, the gradient parameters of the federated learning module cannot be modified once they are uploaded. Therefore, without the authorization of the administrator, an attacker cannot obtain plaintext and modify data. The global model parameters are stored in the distributed file system after being encrypted, and the summary information for each group of data is recorded on the block. And each block contains the time stamp and the hash value of the previous block, it ensures the nontampering of data. We also use the lightweight authentication and homomorphic encryption algorithm [37]. If users do not leak their private keys, even if the message returned from the server is intercepted by malicious users, the meanings cannot be inferred. It protects the user's privacy effectively.

## 5. Conclusions

In this paper, we innovatively propose a blockchain-based federated learning pool framework for data silos and data privacy disclosure in IoV and design a lightweight encryption algorithm called CPC to combine with it. In our framework, the federated learning pool module can select the appropriate federated learning methods according to the distribution of data sources, which makes building the model more accurate and faster. Meanwhile, the application of the CPC lightweight encryption algorithm further ensures the security of data interaction between vehicles. Besides, we

make the blockchain as the bottom layer to establish a trusted mechanism, which ensures the reliability of data transmission. Then, we made an experiment to verify our framework, and the results show our framework is more efficient, safe, and accurate.

At present, our framework is not perfect enough. In the future, we will improve the computational efficiency and accuracy of the federated learning pool and improve the federated learning methods in federated learning pool so that our model can adapt to more scenarios. The CPC lightweight encryption algorithm can reduce the workload of vehicle system. However, as the amount of data rise, the cost will also rise, so we will mainly improve our CPC lightweight encryption algorithm in energy consumption and storage cost. In addition, we will also improve the stability of blockchain to make BFLP more efficient and stable.

## Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## References


[1] H. Gao, C. Liu, Y. Li et al., "V2VR: reliable hybrid-network-oriented V2V data transmission and routing considering RSUs and connectivity probability," *IEEE Transactions on Intelligent Transportation Systems, Early Access*, vol. 13, 2020.

[2] C. Li, *Research on Security Mechanism for Information Security Problems of Vehicle Networking*, Beijing Jiaotong University, Beijing, China, 2019.

[3] Z. Liu, *Research on Key Technologies of Data Transmission and Privacy Protection in Telematics*, Beijing University of Posts and Telecommunications, Beijing, China, 2019.

[4] H. Gao, L. Kuang, Y. Yin, B. Guo, and K. Dou, "Mining consuming behaviors with temporal evolution for personalized recommendation in mobile marketing apps," *Mobile Networks and Applications*, vol. 25, no. 4, pp. 1233–1248, 2020.

[5] Q. Yang, Y. Liu, Y. Cheng, Y. Kang, T. Chen, and H. Yu, "Federated learning," *Synthesis Lectures on Artificial Intelligence and Machine Learning*, vol. 13, no. 3, pp. 1–207, 2019.

[6] H. Gao, W. Huang, and Y. Duan, "The cloud-edge-based dynamic reconfiguration to service workflow for mobile ecommerce environments," *ACM Transactions on Internet Technology*, vol. 21, no. 1, pp. 1–23, 2021.

[7] Z. Yan, J. Wicaksana, Z. Wang, X. Yang, and K.-T. Cheng, "Variation-Aware federated learning with multi-source decentralized medical image data," *IEEE Journal of Biomedical and Health Informatics*, 2020.

[8] J. Kang, Z. Xiong, D. Niyato et al., "Incentive design for efficient federated learning in mobile networks: a contract theory approach," in *Proceedings of the 2019 IEEE VTS Asia Pacific Wireless Communications Symposium (APWCS)*, pp. 1–5, Singapore, August 2019.

[9] S. Niknam, H. S. Dhillon, and J. H. Reed, "Federated learning for wireless communications: motivation, opportunities, and challenges," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 46–51, 2020.

[10] Y. Lu, X. Huang, Y. Dai, S. Maharjan, and Y. Zhang, "Federated learning for data privacy preservation in vehicular cyber-physical systems," *IEEE Network*, vol. 34, no. 3, pp. 50–56, 2020.

[11] S. R. Pokhrel and J. Choi, "A decentralized federated learning approach for connected autonomous vehicles," in *Proceedings of the 2020 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, Seoul, South Korea, May 2020.

[12] A. M. Elbir and C. Sinem, "Federated learning for vehicular networks," 2020, http://arxiv.org/abs/2006.01412.

[13] J. ZouariM. Hamdi et al., "A privacy-preserving homomorphic encryption scheme for the internet of things," in *Proceedings of the IEEE 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pp. 1939–1944, Valencia, Spain, June 2017.

[14] M. Salem, S. Taheri, and J.-S Yuan, "Utilizing transfer learning and homomorphic encryption in a privacy preserving and secure biometric recognition system," *Computers*, vol. 8, no. 1, p. 3, 2019.

[15] W. She, Z.-H. Gu, X.-K. Lyu, Q. Liu, Z. Tian, and W. Liu, "Homomorphic consortium blockchain for smart home system sensitive data privacy preserving," *IEEE Access*, vol. 7, pp. 62058–62070, 2019.

[16] D. Zhang, X. Lei, M. Tang, K.-C. Li, and A. Zomaya, "Circuit copyright blockchain: blockchain-based homomorphic encryption for IP circuit protection," *IEEE Transactions on Emerging Topics in Computing*, 2020.

[17] P. Budhwar, "TinyOS: an operating system for wireless sensor networks," *Proceedings of the IJCST*.vol. 8491, 2015.

[18] S. R. Pokhrel and J. Choi, "Federated learning with blockchain for autonomous vehicles: analysis and design challenges," *IEEE Transactions on Communications*, vol. 68, no. 8, pp. 4734–4746, 2020.

[19] D. Das, S. Banerjee, and U. Biswas, "A secure vehicle theft detection framework using Blockchain and smart contract," *Peer-to-Peer Networking and Applications*, vol. 14, pp. 672–686, 2021.

[20] D. B. Rawat et al., "Blockchain enabled named data networking for secure vehicle-to-everything communications," *IEEE Network*, vol. 34, no. 5, 2020.

[21] H. Liu, Y. Zhang, S. Zheng, and Y. Li, "Electric vehicle power trading mechanism based on blockchain and smart contract in V2G network," *IEEE Access*, vol. 7, pp. 160546–160558, 2019.

[22] Y. Liu, J. Peng, J. Kang, Iliyasu et al., "A secure federated learning framework for 5G networks," 2020, http://arxiv.org/abs/2005.05752.

[23] Y. Liu, Y. Kang, X. Zhang et al., "A communication efficient vertical federated learning framework," 2019, http://arxiv.org/abs/1912.11187.

[24] Y. Liu, Y. Kang, C. Xing et al., "A secure federated transfer learning framework," *IEEE Intelligent Systems*, vol. 35, no. 4, pp. 70–82, 2020.

[25] B. McMahan, E. Moore, D. Ramage et al., "Communication-efficient learning of deep networks from decentralized data," in *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics*, pp. 1273–1282, PMLR 54, Lauderdale, FL, USA, April 2017.

[26] K. Cheng, T. Fan, Y. Jin et al., "Secureboost: a lossless federated learning framework," 2019, http://arxiv.org/abs/1901.08755.

[27] X. Ma, H. Gao, H. Xu et al., "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 249, 2019.

[28] X. Yang, S. Zhou, and M. Cao, "An approach to alleviate the sparsity problem of hybrid collaborative filtering based recommendations: the product-attribute perspective from user reviews," *Mobile Networks and Applications*, vol. 25, no. 2, pp. 376–390, 2020.

[29] E. Biham and A. Shamir, "Differential cryptanalysis of DES-like cryptosystems," *Journal of CRYPTOLOGY*, vol. 4, no. 1, pp. 3–72, 1991.

[30] W. Diffie and M. Hellman, "New directions in cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644–654, 1976.

[31] T. M. ChenJ. Blasco et al., "Cryptography in WSNs," *Mission-Oriented Sensor Networks and Systems: Art and Science*, pp. 783–820, Springer, Cham, Switzerland, 2019.

[32] M. Novelan, A. Husein, and M. Harahap, "Sms security system on mobile devices using tiny encryption algorithm," *Journal of Physics: Conference Series*, vol. 1007, no. 4, pp. 12–37, 2018.

[33] S. Kotel, M. Zeghid, and M. Machhout, "Lightweight encryption algorithm based on modifed XTEA for low-resource embedded devices," in *Proceedings of the 21st International Database Engineering and Applications Symposium*, pp. 192–199, Bristol, UK, July 2017.

[34] M. Panda, "Data security in wireless sensor networks via AES algorithm," in *Proceedings of the IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*, pp. 1–5, Coimbatore, India, January 2015.

[35] M. Zhao and Z. Yao, "Cloud computing access control model based on RBAC," *Computer Application*, vol. 32, pp. 267–270, 2003.

[36] E. Yuan and J. Tong, "Attributed based access control (ABAC) for web services," in *Proceedigns of the International Conference on Web Services (ICWS'05)*, IEEE, Orlando, FL, USA, July 2005.

[37] Y. Aono, T. Hayashi, L. Trieu Phong et al., "Scalable and secure logistic regression via homomorphic encryption," in *Proceedings of the Sixth ACM Conference on Data and Application Security and Privacy*, pp. 142–144, New Orleans, LA, USA, March 2016.