Federated Learning based on Homomorphic Encryption and Digital Signatures

Zao Chen *

Department of Computer and information engineering college, Tianjin Normal University, 300387, Tianjin, China

* Corresponding author email: 3180400002@caa.edu.cn

Abstract. Federated learning aims to train a centralized federated model using decentralized data sources and to ensure the security and privacy of user data during system training process. Federal learning still faces four challenges: high communication costs, system heterogeneity, data heterogeneity, and data security, The primary attacks facing federal learning include confidentiality, integrity, and usability attacks. Attackers mainly target confidential attacks, while malicious attackers target integrity attacks and usability attacks. Homomorphism encryption refers to the encryption algorithm that satisfies the nature of the homomorphism operation, that is, to the ciphertext of the data after the homomorphic encryption. This article first introduces the federated learning and its possible attacks in the process of modeling behavior, and then introduces how to use homomorphic encryption and digital signature algorithm to prevent the attack method, finally through the experimental reality of federated learning encryption and digital signature and analyzed the above operation on the performance of the federated learning system.

Keywords: Federated Learning; Homomorphic Encryption; Digital Signatures; HASH Function; Encryption Loss.

1. Introduction

In 2016, in order to solve their android mobile phone user update model, protect mobile phone terminal data and personal data privacy, ensure that user information leakage, ensure that multiple participants or more effective machine learning large data exchange between participants or computing nodes, Google created the federal learning, it is a kind of emerging artificial intelligence technology. Among them, machine learning algorithms for federated learning are suitable not only for neural network computing, but also for basic algorithms such as random forests. Predictive federated learning is expected to become the foundation of the next generation of artificial intelligence collaborative computing and network security communication [1].

Federal learning has the following advantages:1) Data isolation can be conducted, and user data participating in machine learning will not be leaked outside, and user privacy protection and data security are guaranteed; 2) Federated learning ensures the quality of models, ensures that federated models outperform the usual separated independent models, and produce no negative migration; 3) That the status of all participants in machine learning is equal, and they can achieve fair cooperation;4) It can ensure that the participating parties can encrypt and exchange the information and the model parameters while maintaining their independence to ensure the security of the data in the model operation [2].

Federated transfer learning (FmL) is a method of learning that can be applied to different data sets. For different data sets, federated learning is divided into horizontal, vertical, and federated transfer learning. Here is a brief classification of federated learning [3].

(1) Horizontal federal learning. Cross-sectional federation learning is characterized by cross-sectional partitioning, the training data of each participant is divided horizontally, the federation of multi-row samples with the same characteristics from multiple participants. In practical application, it can be understood that when multinational companies (e.g., supermarkets in different countries) cooperate, the user information they share differs. When they jointly train the user model, they find

that the user has the same feature options. Therefore, this is the most efficient time to apply cross-sectional federation learning.

(2) Vertical federal learning. The principle of longitudinal federation learning as the most commonly used federation learning approach is that we can slice and dice the existing dataset in longitudinal dimensions when the users of two datasets overlap more and the user features overlap less. Longitudinal federation learning, also known as sample-aligned federation learning, finds the common sample IDs of the participants before the server trains the gradient data submitted by multiple parties, a process also called "database crashing." In a practical application, for example, a bank and an e-commerce platform in the same region (at this point, the target users can be said to overlap). Each party of the two platforms can input X and Y feature values in their own hands for local training. The bank has the local can be user's income and expenditure records x1, and the e-commerce platform has the user's consumption records and browsing records x2. Based on these data, the local user's behavior and credit value Y are rated, extracting different feature descriptions of the same target from multiple parties for training. Both parties can take different measures for users based on the aggregated model data, such as whether to continue pushing loan ads and luxury ads or stop lending to the target users while keeping all their confidential information undisclosed.

(3) Federal migration learning. Instance-based Federated Transfer Learning: For cross-sectional federation learning, the participants' data often came from different regions, which would potentially lead to poor performance of the machine learning models trained on these data [2]. Participants may selectively pick or weight-training samples to reduce distribution differences so that the target loss function can be minimized. For longitudinal federation learning, the participants may have very different business goals. Therefore, aligned samples and some features may negatively affect federated migration learning, referred to as negative migration. In this case, the participants can selectively choose the features and samples used for training to avoid negative migration. Feature-based federated transfer learning: Participants collaborate to learn a common representation space. The semantic differences and distribution between the representations converted from the original data can be alleviated in this space, so that the knowledge can be transferred between the domains. For horizontal federation learning, a common representation space can be learned by minimizing the maximum average difference between the samples of participants. For longitudinal federation learning, the common representation space can be learned by minimizing the distance between the representations belonging to different participants in the aligned samples. Model-based Federated Migration Learning: The participants collaboratively learn a shared model that can be used for migration learning, or the participants use a pre-trained model as the initial model for all or part of the federation learning task. Horizontal federation learning is a form of model-based federated transfer learning because, in each communication round, the participants collaboratively train a global model. Each participant uses that global model as the initial model for fine-tuning. For longitudinal federation learning, predictive models can be learned from aligned samples or using semi-supervised learning techniques to infer missing features and labels. Then, more accurate shared models can be trained using the expanded training samples.

Three federal learning: horizontal federal learning, longitudinal federal learning, and transfer federal learning are suitable to solve different practical problems. In practical application, different types should be selected according to the actual application scenarios.

2. Federal Learning Architecture

In federal learning, the common goal of the participants is to collaboratively train a federated model while keeping the original data intact. Each participant maintains a local model based on local data during training. At the same time, the server aggregates the local model using various possible aggregation rules and obtains a global model.

The central server selects some clients based on their status (charging, idle, wifi connection status), as shown in Figure 1. The selected client downloads the current model parameters or weights from

the central server and uses these weights to initialize the local model. The client uses local data to optimize the model using algorithm, which is executed multiple times by the client to reduce communication costs [4]. After the client training, the optimized training parameters are sent to the central server. However, due to the instability of the network, the client may be poor connection, computing resources are not enough to complete the calculation, the amount of training data is too large, and during the training period, some may withdraw in the parameter transmission stage. At this point, the ratio of failed clients that the central server cannot handle is reported, and the process continues based on the number of updates received. If the number of customers reported in time does not reach the predetermined number, this round of update will be abandoned, and the next update process will be renewed. The central server that weights the customer according to its data set size aggregates updates from all clients, produce a new shared model.

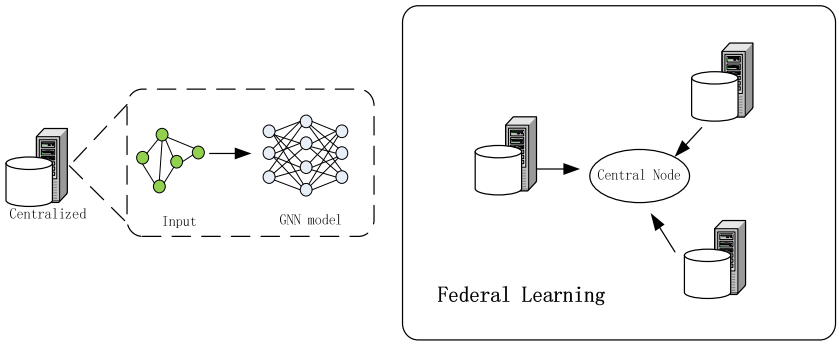

Figure 1. Federal Learning Architecture

3. Federal Learning Challenges and Characteristics

Federated learning is not typical of distributed learning, primarily due to the following challenges and characteristics [5].

(1) Non-IID data: The data distribution is different for different clients and there may be dependencies between them.

(2) Unbalanced data: there is a gap in the amount of data from different clients

(3) Large-scale distribution of data: large number of clients and sometimes even greater than the average number of samples per client.

(4) Unreliable device connections: In most cases, client connections are slow, limited, expensive and unavailable, significantly reducing the number of available connections. In some cases, among the clients involved in the training, many clients may not be able to complete each round of learning due to poor computing power.

(5) Limited device memory: there will be many IoT/mobile devices involved in the training, and these devices have limited memory

(6) Poisoning attack: Since the client is anonymous, the attacker will behave like a normal user and join the FL in the selected system. Therefore, an attacker can prevent the model from achieving the prediction or prediction outcome by adding toxic data to the training process.

4. The Main Attacks on Federal Learning

Evasion Attack (EA): It refers to the purpose of spoofing the model by modifying the input samples without changing the machine learning model by the attacker. Evasion attack mainly occurs in the model inference phase, a common attack pattern in federal learning and traditional centralized training scenarios.

Data Poisoning Attack: Also known as data poisoning. Machine learning models are trained based on historical sample data, so an attacker can tamper with the training data to make the trained model output as the attacker intends. Data attack is a pervasive attack mode in federation learning scenarios.

Since each device side involved in federation training is independent of the other, when a client is held hostage, the attacker can take full control of the client, including tampering with its local data, to achieve the purpose of contaminating the whole global model, and the backdoor attack is a typical data attack scheme.

Model Attack: A model attack refers to an attacker who modifies the parameters of a model to destroy the model during the training process. In traditional centralized training, the model's training process is obtained by the user inputting data first and then iteratively trained by gradient descent and other optimization methods. Hence, there is usually no way for the user to participate in the intermediate training process, and it is difficult to tamper with the model during the training process. However, in the scenario of federation learning, the training process of federation learning, the model will be transmitted between the client and the server for several interactions. As with data attacks, when the attacker holds a client hostage, then the attacker can also modify the acquired global model and upload the modified model to the server to achieve the purpose of the attack.

Model Inversion Attack: It refers to the attack to obtain the model's parameters or the raw data for training by performing specific reverse engineering on the model, including Model Extraction Attacks, Membership Inference Attacks, and Model Inversion Attacks).

5. Defense of Federal Learning

The protection of federal learning is mainly provided in the following several ways:

Homomorphic encryption: From the perspective of computation, due to the nature of homomorphic encryption, the computation result of data in the encrypted state is the same as the computation result in the plaintext state. From the security point of view, due to the data encryption, even if the model is stolen, the attacker cannot know the real parameters of the model and can effectively prevent the attack on the model. Considering the balance between security and efficiency, the semi-homomomorphic encryption algorithm is often used in practical applications.

Differential privacy: Differential privacy is a widespread secure computing scheme. Unlike homomorphic encryption, differential privacy protects model parameters and data privacy security by adding noise [6].

Model compression: Model compression makes models lighter for easy deployment and transmission. In addition, the model is compressed so that the user only gets partial information about the model's parameters, preventing leakage of the original model.

Parameter sparse: Parameter sparse is also an implementation of model compression. By combining the mask matrix, only part of the parameters is transmitted so that even if the model is stolen, it is difficult for the attacker to restore the original model, thus achieving the purpose of protecting it [7].

Anomaly detection: A more practical solution for data poisoning and model tampering is to detect abnormal client models through anomaly detection. In addition, the selection mechanism of federal learning can prevent malicious model continuity attacks to a certain extent.

6. Homomorphic Encryption

In 1978, the concept of homomorphism encryption (HE) was first proposed by Rivest et al [8]. Homomorphic encryption provides a function for processing encrypted data and is an encryption technique that allows computational operations to be performed on the ciphertext and generates an encrypted result. The computation result obtained on the ciphertext is decrypted to match the result on the plaintext as if the same computation operation was performed on the plaintext, as shown in Figure 2.

Due to performance constraints and other factors, semi-homomorphic encryption algorithms are currently used mainly in industry. Which implements Logistic Regression training in the encrypted state with semi-homomomorphic encryption as a security mechanism under federal learning. Paillier

semi-homomomorphic encryption algorithm was proposed by Pascal Paillier in 1999 and it is an additive semi-homomomorphic encryption.

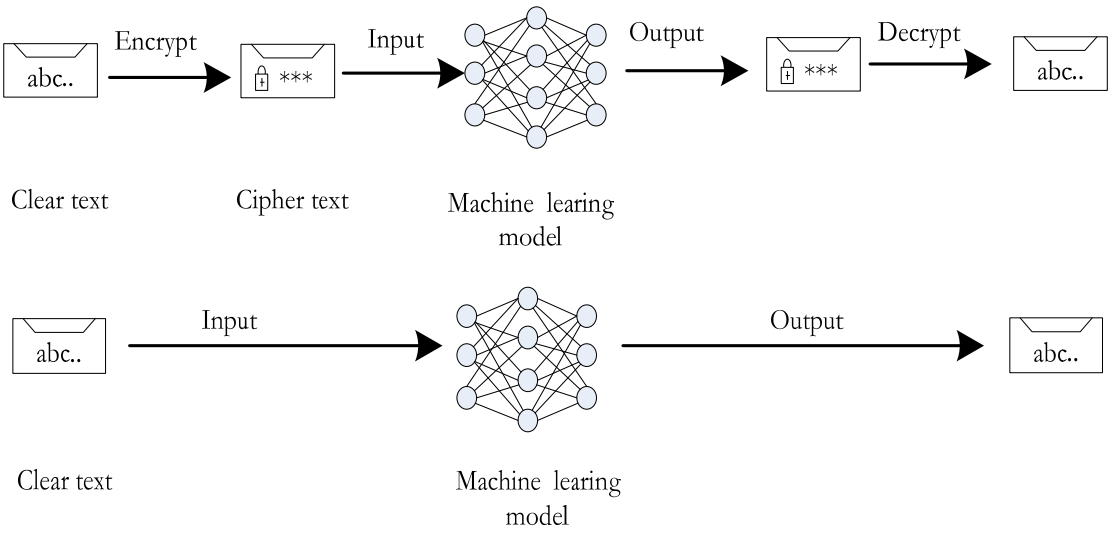

Figure 2. Homomorphic encryption workflow

If user uses u for the plaintext and $[[u]]$ for the ciphertext, the Paillier semi-homomomorphic encryption algorithm satisfies: $[[u + v]] = [[u]] + [[v]]$.

7. Encryption Loss Function

Suppose that there are currently n sample datasets, set as:

$$T = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n) \tag{1}$$

The logarithmic loss function of LR can be written as:

$$L = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i \theta^T x_i}\right) \tag{2}$$

The above equation is derived to obtain the gradient of the loss function value L about the model parameter θ, $\frac{\partial L}{\partial \theta}$ which is satisfied:

$$\frac{\partial L}{\partial \theta} = \frac{1}{n}\sum_{i=1}^{n} \left(\frac{1}{1+e^{y_i \theta^T x_i}} - 1\right) y_i \, x_i \tag{3}$$

Using gradient descent, it can find the parameter update for each step:

$$\theta = \theta - lr * \frac{\partial L}{\partial \theta} \tag{4}$$

The above calculation process, including the parameters θ and data (x, y) are calculated in plain text, in the federated learning scenario, this practice will be the risk of data leaks, and the federal learning based on HE, requires in the encrypted state of parameters, that is, the transmission parameter θ is an encrypted value $[[\theta]]$, the loss function can be rewritten as:

$$L = \frac{1}{n}\sum_{i=1}^{n} \log\left(1 + e^{-y_i [[\theta]]^T x_i}\right) \tag{5}$$

The above equation involves exponential operation and logarithmic operation of encrypted data, but we have explained earlier that Paillier encryption algorithm only supports additive homomorphism and scalar multiplicative homorphism, not multiplication homomorphism, nor complex exponential and logarithmic operation, so it cannot be solved in the encrypted state. Previous works proposed a method of Taylor loss to approximate the loss of the original logarithmic loss, namely by the Taylor expansion of the original logarithmic loss function, the polynomial to approximate the log loss function by polynomial, after the Taylor expansion, the loss function into only scalar multiplication and addition operations, which can be directly applied to Paillier to perform the encryption solution [1].

For any function f (x), its Taylor polynomial expansion at x=0 can be expressed as:

$$f(x) = \sum_{i=0}^{\infty} \frac{f'(0)}{i!} x^i \tag{6}$$

When $f(z)$ is a logarithmic loss function, namely the Taylor expansion expression $f(z) = \log(1 + e^{-z})$ at z = 0:

$$\log(1 + e^{-z}) \approx log2 - \frac{1}{2}z + \frac{1}{8}z^2 + o(z^2) \tag{7}$$

Here we use the second order polynomial therein to approximate the logarithmic loss function $z = y[[\theta]]^T x$ and substitute it into the above formula, obtaining:

$$\log\left(1 + e^{-y\theta^T x}\right) \approx log2 - \frac{1}{2}y_i\theta^T x_i + \frac{1}{8}(\theta^T x_i)^2 \tag{8}$$

The last of these, since $y^2 = 1$, it directly removes y and substitute (8) into (2):

$$L = \frac{1}{n}\sum_{i=1}^{n}\{log2 - \frac{1}{2}y_i\theta^T x_i + \frac{1}{8}(\theta^T x_i)^2\} \tag{9}$$

For equation (9), the loss value L and the gradient θ value is:

$$\frac{\partial L}{\partial \theta} = \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{4}\theta^T x_i - \frac{1}{2}y_i)x_i \tag{10}$$

The corresponding encryption gradient of the resulting formula (10) is:

$$[\left[\frac{\partial L}{\partial \theta}\right]] = \frac{1}{n}\sum_{i=1}^{n}(\frac{1}{4}[[\theta]]^T x_i - \frac{1}{2}y_i)x_i \tag{11}$$

8. Experimental Results and Analysis

An open-source project (launched by Microsoft's AI division) based on the FATE framework is designed to provide a reliable and secure computing framework for the federated learning ecosystem. The model's accuracy after training with federal learning under homomorphic encryption using the training data from the breast cancer dataset (Breast cancer) is shown in Figure 3.

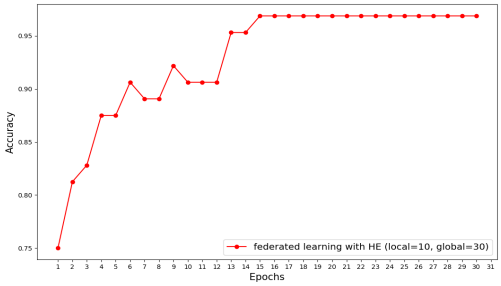

Figure 3. 30 total iterations yielded

Ten local iterations and 30 total iterations yielded a relatively high accuracy after the 15th.

8.1 Digital Signature

Digital signature technology is to attach a piece of validation data when sending and receiving the data, the sender uses the self-generated private key, through a specific HASH calculation, the summary information obtained from the original information is encrypted, and the summary is sent to the receiver along with the original text [9]. The receiver has the sender's public key, the receiver decrypts the public key to generate a new summary information and extracts a received summary information with the HASH function, comparing the above two decrypted summaries. Even if one byte in the data changes, the HSA function produces different summary. If the alignment results are inconsistent, if the information received is modified, the data is discarded; otherwise, the data is not modified during transmission and is correct and complete. So, the digital signatures are able to verify that the source of the information is correct [10].

An important means to ensure the secure realization of data communication between the nodes in the network, the digital signature mechanism can solve the problems of information impersonation, data falsification, Information denial and data tampering existing in the process of data transmission. People use digital signatures to replace manual signatures and seals previously used in network environments, acting as follows:

(1) Anti-counterfeiting (forgery), the private key is only stored by the signature, others are impossible to construct the private key.

(2) The identity of the sender can be identified. In the process of data transmission, the receiver can identify the authenticity of the identity of the sender.

(3) Prevent information damage and tamper prevention to ensure the integrity of the information.

The working procedure is shown in Figure 4.

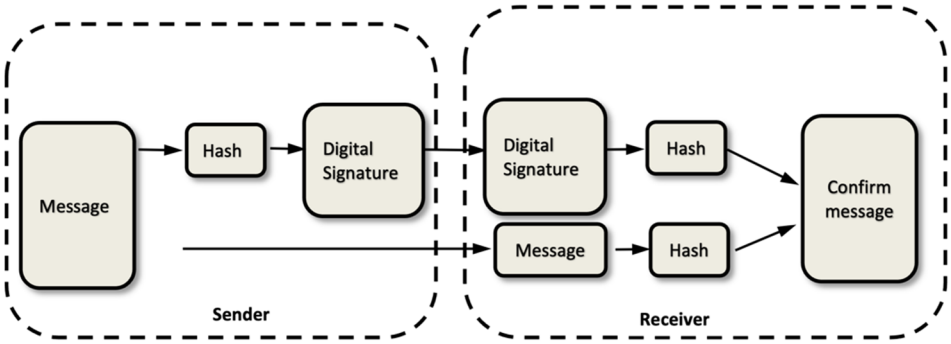

Figure 4. Digital Signature and verification

8.2 Digital Signature Homomorphic Encryption and Digital Signature on Federal Learning

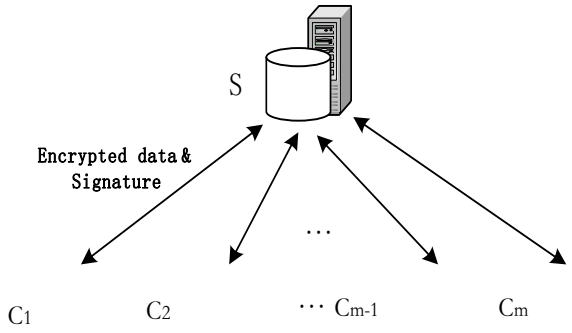

Figure 5. Homomorphic Encryption and Digital Signature on Federal Learning

SM2 elliptic curve public key cryptography algorithm: China's independent intellectual property rights of commercial cryptographic algorithms is a kind of ECC (Elliptic Curve Cryptosystem)

algorithm based on the elliptic curve discrete logarithm problem, the computational complexity is exponential, the solution is more difficult, the same level of security requirements, elliptic curve cryptography than other public key seconds and hair required key length is smaller A lot.

Abstract algorithm (signature algorithm) international SHA-256 and national secret SM3:

The SM3 algorithm is an improved implementation of SHA-256 for digital signatures and authentication in commercial cryptographic applications. The SM3 algorithm uses a Merkle-Damgard structure with a message group length of 512 bits and a digest value of 256.

8.3 Analysis

The impact of homomorphic encryption on the performance of federal systems. Thirty iterations were performed, and a relatively high accuracy was also obtained after the 16th iteration. The test results after using homomorphic encryption are shown in Figure 6.

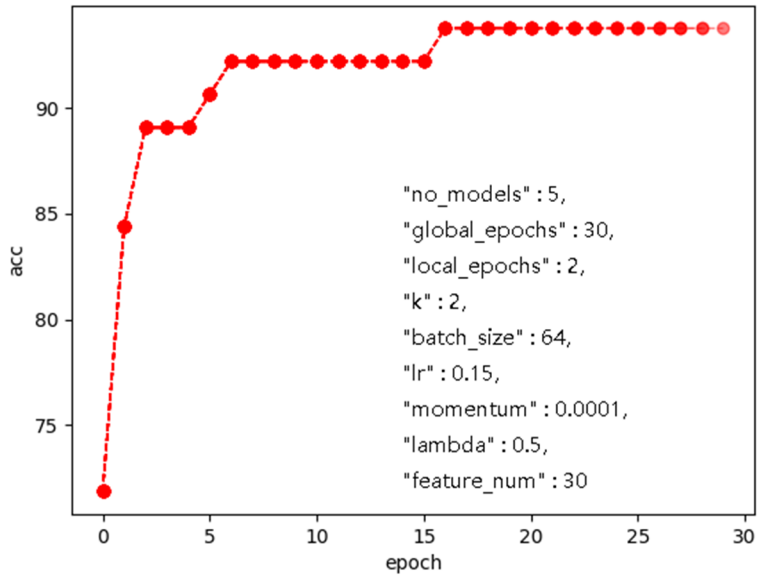

Figure 6. Test results for Homomorphic Encryption on Federal Learning

Impact on federal system performance with the addition of data signatures and signature verification. Paillier encryption and digital signature FedAvg for performance evaluation, select the federated learning environment of both parties involved, under the MNIST data set, the model training iterations, the test results show that: using homomorphic encryption and digital signature do not affect the accuracy of federated learning, but the computational efficiency is reduced.

9. Conclusion

In the federal learning in the case of not leaking user data to user data implementation collaborative encryption calculation, data and the intermediate calculation results in the process will not leak to any party, experiments proved that using homomorphic encryption and digital signature, although the impact of machine learning efficiency, but can greatly improve the integrity of user data security in the federal learning. In the future, we will work to optimize the federated algorithms for encryption and digital signatures to improve the operational efficiency.

References

[1] JianBin Li, YuQi Ren, SuWan Fang, KunChang Li, MingYu Sun. "Federated Learning-BasedUltra-Short term load forecasting in power Internet of things", 2020 IEEE International Conference on Energy Internet (ICEI), 2020.

[2] Q. Yang, Y. Liu, Y, Chen, and et al, Federated Learning. Morgan & Claypool Publishers,Dec.2019.

[3] QiangYang, YangLiu, Tianjian Chen, and Yonxing Tong, Federate machine learning: Concept and applications. ACM Trunsactions on Intelligent System and Tachnology,2019.

[4] Tyler B. Johnson, Pulkit Agrawal, Haijie Gu, Carlos Guestrin, AdaScale SGD: A User-Friendly Algorithm for Distributed Training, arXiv: Learning 2020.

[5] Qiang Yang, Anbu Huang, YangLiu, Tianjian Chen, Practicing Federated Learning, Federal Learning and Practice, Electronics Industry Press, ISBN 978-7-121-40792-5.

[6] DWORK C, ROTH A. The algorithmic foundations of differential privacy. Foundations and Trends in Theoretical Computer Sciencde,9(3):211-407,2-14.

[7] Jakub Konecny, H. brendan McMahan, Felix X. Yu, Perter Richtarik, Ananda Therrtha Suresh, and Dave Bacon. Federated learning: Strategies for improving communication efficiency. CoRR, abs/ 1610. 05492. 2016.

[8] Pascal Palillier. Public-key cryptosystems base on composite degree residuosity classcs. In Jacques Stern, editor, Advances in Cryptology--Eeurocrypt'99, PAGES 223-238, Berlin, Heidelberg.1999, Springer Berlin Heidelberg.

[9] SM3 Cryptographic Hash Algorithm, China Standardization Cryptography Industry Technical Committee.

[10] Public Key Cryptographic Algorithm SM2 Based on Elliptic Curves, SM2 elliptic curve public key password algorithm, China Standardization Cryptography Industry Technical Committee.