Data Privacy Protection in Data Fusion

Zhiping Li a, *, Jiagui Xie b, Likun Gao c, Fanjie Nie d
China ICT Institute, No. 52 Huayuan North Road, Haidian District, Beijing, China

* a Corresponding author e-mail: lizhiping@caict.ac.cn, b xiejiagui@caict.ac.cn, c gaolikun@caict.ac.cn, d niefanjie@caict.ac.cn

Abstract. Today, with the rapid development of big data technology, multi data fusion is facing the problem of privacy security, and people pay more and more attention to its data security. Firstly, this paper describes the three most common problems of multi data fusion, and proposes an integrated federated learning for its security, which combines the trusted execution environment (TEE), asymmetric encryption (RSA) and blockchain technology. Trusted execution environment solves the privacy security problem of local client data processing, asymmetric encryption solves the privacy security problem of data sharing and transmission, and blockchain technology solves the privacy security problem of data tampering in centralized server. Finally, the simulation results show that compared with the traditional federated learning, the fusion federated learning has the same test accuracy, smaller training loss and better training accuracy.

Keywords: Trusted execution environment, Asymmetric encryption, DHT blockchain.

1. Introduction
In recent years, artificial intelligence has developed rapidly, and has gradually entered the practical application stage from the invention period a few years ago. In more and more application scenarios, artificial intelligence technology has been applied. However, with the increase of algorithm and computing power and the increasing demand for data scale, how to meet the data privacy protection demand of artificial intelligence model has become an urgent problem to be solved in the development of artificial intelligence technology [1].

In 2016, Google proposed a new artificial intelligence technology [2] for Android mobile terminal users to update the model locally, which is called federated learning (FL). Federated learning refers to the machine learning setting of the cooperative training model by multiple clients under the coordination of a central server. Federated learning can enable multi-party organizations to achieve AI cooperation without sharing data, and ensure the privacy and security of user data [3]. As shown in Figure 1, the traditional federated learning is implemented under the coordination of the central server. Each participant (local device) only uses its own locally owned data to train the machine learning model and obtain model parameter updates, and sends model parameter updates to a central server (Coordinator), such as model weights or gradient information [4]; The central server aggregates the received model parameter updates from different devices (for example, weighted average), and distributes the
aggregated model parameter updates to local devices. The local devices update the local model with the aggregated parameters [5].

Currently, federated learning is divided into horizontal federated learning, vertical federated learning and federated transfer learning according to the difference of data feature space and sample space [6]. Horizontal federated learning is mainly suitable for data sharing the same sample space but different feature space, while vertical federated learning is opposite to horizontal federated learning. Federated transfer learning is aimed at different feature space and sample space. At present, federated learning has begun to explore the landing in various industries, such as the risk control model of joint modeling of multiple banking institutions in the financial field, effectively improving the matching efficiency of information and resources in the field of smart retail, and realizing cross regional cooperation in the field of health care [7]. Federated learning is also applied in the field of multi-source data fusion. Because the data features are the same and the sample data come from different institutions, we adopt horizontal federated learning. The steps are as follows:

1) Each participant downloads the latest model from the central server.
2) Each participant trains the model with local data, encrypts the gradient and uploads it to the central server, which aggregates the gradient update model parameters of each user.
3) The central server returns the updated model to each participant
4) Each participant updates its own model.

Figure 1. Traditional federated learning framework.

In the traditional machine learning modeling, the data needed for model training is usually collected into a data center, and then the model is trained and predicted [8]. In horizontal federated learning, it can be regarded as a distributed model training based on samples, which distributes all data to different machines. Each machine downloads the model from the server, then trains the model with local data, and then returns the parameters to be updated by the server; The server aggregates the returned parameters of each machine, updates the model, and feeds back the latest model to each machine. In this process, each machine has the same and complete model, and there is no communication and dependence between the machines. Each machine can also predict independently in the prediction process. This process can be regarded as a distributed model training based on samples. Google initially used horizontal Federation to solve the problem of Android mobile terminal users updating the model locally.

2. Privacy Security Problems in Data Fusion

2.1. Privacy Issues in Local Execution Environment
Local execution environment privacy issues. All mobile devices support rich execution environment (REE) [9], run common OS: Android, IOS, Linux, provide all the functions of the device for the upper app, open, scalable and universal, and operate in the interconnected network world. There are security risks in REE: app isolation based on OS is very easy to be bypassed, OS code is huge, vulnerabilities
occur frequently, OS is difficult to be verified and authenticated, OS can see all the data inside app, a large number of malicious code and advanced attack technology, lack of isolation means that app can not safely store key [10].

2.2. Privacy Issues in Data Transmission
In the traditional multi data fusion processing, each local data sharing can improve the service performance trend. Although it provides traversal, it also has potential privacy problems. The terminal equipment must transmit the local data to the central server to manage the data and optimize the local model. However, the deleted data usually contains some unique identifier, location information and moving track, which may lead to further privacy disclosure and property loss [11].

2.3. Data Tampering in Central Server
The central server has security risks brought by centralization. Its mode and data are stored in the server, which can be modified arbitrarily, and can control the storage and deletion of data. There is the possibility of data leakage or capital loss. Centralized servers also need certain development costs and regular maintenance and operation costs, which may cause privacy leakage in the process of operation and maintenance [12].

3. Integrated Federated Learning
The fusion federated learning is a security and privacy protection system based on the combination of trusted execution environment (TEE) and federated learning update model parameters, as well as blockchain storage data, which is suitable for multi data fusion processing

3.1. Trusted Execution Environment
Trusted execution environment, which can ensure that the computing is not interfered by the conventional operating system, is called "trusted". This is achieved by creating a small operating system that can run independently in the "security world" of TrustZone, which directly provides a few services in the form of system calls (directly handled by the TrustZone kernel) [13]. In addition, the TrustZone kernel can safely load and execute the small program "trustlets" in order to add the "trusted" function in the extended model. Trustlets programs can provide secure services for insecure (ordinary world) operating systems, such as Android. TEE is usually used to run critical operations:

1) Mobile payment: fingerprint verification, pin code input, etc.
2) Confidential data: secure storage of private key, certificate, etc.
3) Contents include: DRM (digital rights protection), etc.

TEE has become the standard configuration of biometric devices: use TEE to isolate fingerprint acquisition, storage, verification and other processes. Even if the mobile phone is jailbroken or root, the attacker will not be able to obtain fingerprint data. TEE runs a complete operating system internally, and runs in isolation from REE (such as Android). TEE and REE interact through shared memory: between OS / between applications [14]. TEE is also divided into kernel mode and user mode. The user mode of TEE can run many different security applications (TA).

Figure 2. Execution environment of REE and TEE.
TEE is corresponding to REE (rich execution environment), which is generally called secure world and normal world [15]. Linux runs on the normal world, but some behaviors with high security requirements, such as fingerprint comparison and private key signature during payment, need to be put into secure world. As shown in Figure 2, TEE has its own execution space, that is to say, there should be an operating system in TEE environment. The security level of TEE is higher than that of rich OS, but lower than that of Se (smart card); On the other hand, the cost of adding TEE is lower, but the cost of se is higher. The hardware and software resources that TEE can access are separated from rich OS. TEE provides a secure execution environment for authorized security software (trustapp trusted application, referred to as TA) and protects the confidentiality [16], integrity and access rights of TA's resources and data. In order to ensure the trusted root of TEE, TEE should be verified and isolated from rich OS in the process of safe startup. In TEE, each TA is independent of each other and cannot access each other without authorization. In short, there are also corresponding application programs (TA) on the operating system of TEE environment. In addition to the independence between the running environment of TEE and the ordinary operating system, each TA in TEE also needs to be authorized and run independently.

3.2. Blockchain
Blockchain is the core supporting technology of digital cryptocurrency system represented by bitcoin, which has the characteristics of decentralization, distrust, flexibility and security. The core advantage of blockchain lies in decentralization, that is, blockchain technology does not use the central organization to establish the trust relationship between distributed nodes, it stores the data in the network nodes and updates them in real time. All nodes in the blockchain network participate in the maintenance together, and connect the blocks to form a data chain according to the sequence of block generation time. As long as not all the participating nodes in the network collapse at the same time, it can run all the time. In the past two years, blockchain has gradually separated from bitcoin and become a hot spot of technological innovation. It has created a new distributed data storage technology with multi node independent participation, which has attracted great attention of government departments, financial institutions and capital markets.

Although blockchain technology is born of digital currency, it has been paid more and more attention in other fields, such as health care, transportation, audit, archives, retail and other people's livelihood fields as well as intelligent applications, because of its decentralized design and excellent security functions. But at the same time, blockchain technology in the field of expansion also encountered some problems to be solved, such as block capacity limitation, long confirmation time, large energy consumption of consensus mechanism based on workload proof, 51% attack and so on. These problems also limited the large-scale commercial application of blockchain.

3.3. Integrated Federated Learning
The traditional federated learning model, which is trained and uploaded by each device locally, updates the model through a central server, so that the user data is only processed on the local device, so as to ensure the user's data privacy. The limitation of this model is that it only depends on a single central server and is vulnerable to server failure. At the same time, there is no suitable incentive base to encourage users to provide data training and upload model parameters. In order to solve the above problems, the invention proposes a blockchain federated learning (also known as integrated FL) based on the trusted execution environment TEE, which uses the blockchain network to replace the central server. The blockchain network allows the local model update of the switching device, and verifies and provides the corresponding incentive mechanism.

The logical structure of integrated FL is composed of devices and miners, as shown in Figure 3. Miners are physically randomly selected devices or individual nodes. Each device calculates and uploads local model updates to the associated miners in the integrated federated learning integrated FL network. The miners exchange and verify all local model updates, and then run the workload proof mechanism pow. When the miner completes the pow, a new block will be produced, in which the local model
updates are recorded. Then, the block storing the local aggregation model updates is added to the blockchain, and then downloaded by each device, and the device calculates the global model updates from the new block. Global model update on local device can ensure that global model update of other devices will not be affected in case of miner or device failure. In this way, the general equipment users and miners users can get the latest global update model, so as to form an incentive mechanism for users. At the same time, the author also analyzes the delay of blockfl caused by blockchain network. In order to increase the practicability of the system, we consider minimizing the delay by adjusting the generation rate of blocks, that is, the difficulty of POW.

Figure 3. Logical structure diagram of integrative federated learning.

3.4. Analysis of Federated Learning

The flow chart of converged federated learning in this paper is shown in Figure 4. The TEE trusted execution environment is safe and controllable. The local client adopts Intel SGX. Intel SGX (Intel Software guard extensions) is the TEE implementation provided by Intel. Due to Intel's mainstream position in the market, Intel SGX has naturally become one of the most commonly used TEE solutions. In SGX, the TEE environment for executing code is called enclave, and the data in enclave can ensure its confidentiality and integrity; Intel also provides a to verify whether a result is executed by SGX, so as to ensure that the malicious person disguises as SGX to steal information; Finally, the SGX scheme provides a broader security boundary. Memory data, including BIOS, cannot obtain the encrypted data in SGX.

Figure 4. Flow chart of integrative federated learning.
RSA encryption technology is more efficient. When the data of the local client is transmitted to the server, only parameters and gradients are transmitted, which is more efficient than the efficiency of transmitting all local data to the server. Firstly, the receiving node generates a pair of keys, namely private key and public key; Then, the receiving node sends the public key to the transmitting node; Secondly, the sending node encrypts the data with the received public key, and then sends it to the receiving node; Finally, after receiving the data, the receiving node decrypts it with its own private key.

DHT blockchain ensures the security of Federated learning. The first is in the blockchain, which is tamper proof. This requires a network large enough to prevent untrusted terminal devices from taking over it. The second is that block storage can properly protect their keys during operation, for example, using some security services from the decentralized server. From here, we will explain how the system protects opponents from tampering with the data stored in the blockchain network. In this case, we seldom consider the adversary modifying the agreement or stealing personal sensitive information. In this model, only the decentralized server has full control over the sensitive data. It is difficult for a malicious party to disguise as a terminal device or destroy the whole network, because the blockchain is completely decentralized. In addition, model updates require digital signatures. Therefore, we believe that malicious parties cannot forge digital signatures or control most networks (more than 50%). In addition, malicious parties cannot poison data because the data is stored outside the chain rather than in public accounts. There is only pointer information encrypted with hash function in public ledger. Even if we consider that the malicious party controls one or some nodes in the distributed hash table network, we can't know any information about the data to the malicious party. The basic principle behind this is that the data is encrypted with a key that cannot be accessed by other nodes. The worst result is that the malicious party is authorized and destroys several local copies of the data, but the system can still recover it because there are a large number of copies distributed throughout the network. Finally, the mixed identity mechanism ensures that the possibility of data poisoning is minimal, because it requires obtaining the signature key and encryption decryption key. If a malicious party happens to steal one of the keys, the sensitive data is still secure.

Analyze the flow chart of integrated federated learning shown in Figure 4:

1) TEE trusted execution environment solves the data privacy problem of local execution environment. The local client partition TEE, in this trusted execution environment area for federated learning training model, and upload parameters.

2) RSA encryption solves the problem of data privacy when sharing data. In the process of federation learning, the model parameters are passed, but the user information data of the client is not passed. Miner nodes aggregate data parameters, RSA generates public key and private key, public key encrypts data, private key decrypts data.

3) Blockchain solves the data tampering problem of centralized server. All participating storage parameter nodes form a decentralized block chain storage system based on DHT protocol, in which miner nodes generate a new global model parameter of block storage through workload proof mechanism pow.

4) The local client downloads the global model parameters and returns to step 1 for iteration.

4. Experimental Simulation

We compared two federated learning methods in the experiment. The first is the traditional federated learning method, that is, the data sets of all client devices are trained, and then the parameters are uploaded to the central server for retraining and callback iterative training according to the model. The second is the integrated federated learning method, which combines the TEE trusted execution environment and blockchain data storage of the client.

Combined with the classical CNN as its task model. In particular, integrated federated learning includes six 3 * 3 convolution layers, including 2 * 32, 2 * 64 and 2 * 128 channels. Re * 2 is activated in each channel pool and maximized by re * 2. In addition, behind the six channels are three fully connected layers, including 382 and 192 relu activated units and another 10 soft Max activated units. In this case, the model will have about 3.6 million fashion Minist parameters and 4.6 million cifar-10
parameters. The block size is allocated from 14.4 MB to 18.3 MB, and the data type is set to 32-bit floating point. Although some other depth models will have better performance, they are not the focus of this model and are not considered. Update the super parameter initialization of the global model as follows. The small batch is 40. The number of cycles per round is 40. The initial learning rate of random gradient descent update is 0.30. The learning rate attenuation is 0.95. We simply model the computing power of each terminal device as how many data samples it can train to further update the global model.

- Figure 5 Comparison of training accuracy of Federated learning model; the figure shows that the training accuracy of traditional federated learning is higher than that of fusion federated learning before 500 minutes, and the training accuracy of fusion federated learning is more stable than that of traditional federated learning.

- Figure 6 is a comparison of federated learning and training losses; it can be seen from the figure that the training loss of traditional federated learning is lower than that of integrated federated learning before 175 min, and both of them decrease gradually. After that, the training loss of integrated federated learning becomes less, and the two kinds of federated learning tend to remain unchanged.

- Figure 7 Comparison of test accuracy of Federated learning. As can be seen from the figure, the test accuracy of traditional federated learning is slightly higher than that of fusion federated learning before 100 min, and both of them increase gradually. Between 100 min and 500 min, the accuracy of the two federated learning modes fluctuates, and after 500 min, the accuracy of the two federated learning modes becomes the same.
Acknowledgments
Finally, we would like to thank the Ministry of industry and information technology for the financial support of the special project "network identification construction project" of "2020 industrial Internet innovation and development project".

References
[1] Lim B, Yao M, Liu V, et al. Data Privacy: Mining Big Data for Artificial Intelligence. 2020.
[2] Oza P, Patel V M. Federated Learning-based Active Authentication on Mobile Devices [J]. 2021.
[3] Blanco-Justicia A, Domingo-Ferrer J, S Martinez, et al. Achieving Security and Privacy in Federated Learning Systems: Survey, Research Challenges and Future Directions [J]. 2020.
[4] Chou L, Liu Z, Wang Z, et al. Efficient and Less Centralized Federated Learning. 2021.
[5] Wu F, He S, Yang Y, et al. On the Convergence of Quantized Parallel Restarted SGD for Central Server Free Distributed Training [J]. 2020.
[6] Zhang W, Li X. Federated Transfer Learning for Intelligent Fault Diagnostics Using Deep Adversarial Networks with Data Privacy [J]. IEEE/ASME Transactions on Mechatronics, 2021, PP (99):1-1.
[7] Grama M, Musat M, L Muoz-González, et al. Robust Aggregation for Adaptive Privacy Preserving Federated Learning in Healthcare[J]. 2020.
[8] Nasser A N, Alsabbagh A, H Géza. The effect of changing training data on a fixed deep learning detection model [J]. IOP Conference Series Materials Science and Engineering, 2019, 568:012087.
[9] Li X, Liu A, Xie M, et al. Adaptive Aggregation Routing to Reduce Delay for Multi-Layer Wireless Sensor Networks[J]. Sensors, 2018, 18(4):1216.
[10] Li D, Lin R, Tang L, et al. SGXPool: Improving the performance of enclave creation in the cloud [J]. Transactions on Emerging Telecommunications Technologies, 2019.
[11] Lu Y, Huang X, Dai Y, et al. Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT. IEEE, 2020.
[12] Zhu L, Liu X, Yu L, et al. Blockchain-Enabled Privacy-Preserving Location Sharing Scheme for LBSNs [J]. Mobile Information Systems, 2021, 2021(5):1-15.
[13] Wang Z, Zuo M, Yao S, et al. Internet of Vehicles Based on TrustZone and Optimized RSA [J]. IOP Conference Series Materials Science and Engineering, 2020, 782:022073.
[14] Jang J, Kang B B. Securing a communication channel for the trusted execution environment [J]. Computers & Security, 2019, 83:79-92.
[15] Kornaros G, Bakoyiannis D, Tomoutzoglou O, et al. TrustNet: Ensuring Normal-world and
Trusted-world CAN-bus Networking[C]// 2019 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm). IEEE, 2019.

[16] Cai Y, Wang Y, Lei L, et al. ALTEE: Constructing Trustworthy Execution Environment for Mobile App Dynamically[C]// 2019 IEEE Symposium on Computers and Communications (ISCC). IEEE, 2019.