Research on Blockchain Privacy Protection Mechanism in Financial Transaction Services Based on Zero-Knowledge Proof and Federal Learning

Maoguang Wang, Tianming Wang, Haoyue Ji

Research on blockchain privacy protection mechanism in financial transaction services based on zero-knowledge proof and federal learning

Wang Maoguang[1] Wang Tianming[1] Ji Haoyue[1]

[1] College of Information, Central University of Finance and Economics, Beijing 10000, China

Abstract. In the financial transaction system centering on blockchain technology, institutions at different levels have different powers and roles, so that they have dissimilar private contents to protect. Taking supply chain financing as an example, a multi-level blockchain system is proposed in this paper. The main steps of building the system are as follows: Firstly, commercial banks and regulatory authorities cooperatively establish a risk control model by Federal Learning. Secondly, the private transaction information will be preserved by zero-knowledge proofs for downstream suppliers. Finally, an architecture of multi-level blockchain is designed to supervise the financial trading for guaranteeing credibility. The experimental results show that the system is more beneficial to privacy protection. By incorporating Federal Learning, it can provide stronger security and more reliable risk control. Further, that can also improve the efficiency and performance of the financial transaction system.

Keywords: Blockchain, Federal learning, Zero-knowledge proof, Privacy protection, Supply chain.

1 Introduction

With the global development of blockchain represented by cryptocurrency, blockchain has received extensive attention from the Internet industry, financial institutions, and academia. Blockchain 3.0 has penetrated various industries. The essence of blockchain reduces the cost of trust. As a result, Internet giants and international financial institutions have started to invest heavily in blockchain. Issues such as consensus mechanism and privacy protection continue to be popular topics of current research.

Blockchain is widely used in the financial industry because of its characteristics such as non-tamperability. These financial applications based on blockchain have different functions so that they face different privacy-preserving issues. In this paper, we take the privacy protection of accounts receivable financing in the supply chain as an entry point, solving two problems. One is that how to preserve the privacy of transaction information between suppliers. Another is that how to train the risk control model when the lending institutions such as commercial banks store their private data on local devices.

Firstly, to prevent counterparties from stealing trade secrets in transaction information, it is necessary to establish a transaction mechanism for privacy protection. One way to enhance privacy protection is to generate many random account addresses for each user [1][2] and every address only be used once. However, it is troublesome for the user to maintain a large number of addresses. Therefore, in this paper, we will hide the transaction amount, which could preserve private data while there is only one address/account per user.

Second, a large number of commercial banks apply AI technology to train a more accurate and intelligent risk control model. However, due to the factors such as economic utility, legal policies, and standard systems, data sharing among participants faces the dilemma of "unwillingness, fear, and inability", which forms "data silos" [3], [4]. This has seriously hindered the training and enhancement of risk control models. Therefore, it can not wait to protect the private data of all parties while satisfying the data requirements of risk control models [5]. Federal Learning has the characteristics of "data available but not visible, data not moved but model moved". So we use this technology to achieve collaborative training among participants. The combination of Federated Learning and blockchain ensures a shared ledger smart contracts and collaborative training of model while considering the privacy of data. The main works of this paper consist of three aspects:

1) We design a multi-level blockchain architecture. The architecture is divided into two levels. Especially, commercial banks are viewed as the link between the two levels. The first level is a management blockchain consisting of regulators and commercial banks, which is mainly used to build risk control models and give credit ratings. Another level is the blockchain between commercial banks and suppliers, which can record transaction information in the supply chain.

2) We propose a financial transaction mechanism by incorporating zero-knowledge proof. Particularly, a double balance model is designed to hide the transaction amount, which could protect the private transaction information between suppliers.

3) We propose a secure data-sharing framework that combines blockchain with Federal Learning. Specifically, an algorithm POQ (Proof of Quality) is designed to determine how to select the participants in Federal Learning.

2 Related work

Zero-Knowledge Proof (ZKP) means that a prover can make a verifier believe that a certain assertion is correct without providing any useful information. In recent years, ZKP has been developed rapidly. It is viewed as a very promising solution to protect blockchain privacy. For UTXO (unused transaction output model), the existing privacy-preserving projects are included Zerocash [6], Monero [7], Zerocoin [8], Dash [9], and CoinJoin [10]. However, there are few schemes with effective privacy protection like DSC [11] in the account model. For the blockchain, the most popular zero-knowledge proof system is zk-SNARK. Zk-SNARK have also been used in the literature [12][13] to achieve privacy protection. Among them, the initialized settings of traditional zk-

SNARK are disposable trusted settings for a specific circuit. After the circuit changed, the initialized settings need to be reset [14]. For better reuse of the initialized settings, several new schemes have recently been proposed in academia [15], [16]. Among them, Fractal [17], Halo [18], and Supersonic [19] design the publicly transparent initialized settings that can generate a common reference string (CRS) without additional variables. However, it is still inappropriate to apply these new schemes to blockchains. Because their proof sizes are much larger than these of traditional schemes. Moreover, Sonic [20], Marlin [21] and PLONK [22] design the generic initialized settings that could create a shared and updatable structured reference string (SRS) using additional variables. Therefore, they theoretically support an unlimited number of arbitrary circuits. Besides, Bulletproof[23] is also an effective zero-knowledge proof that does not require the trusted initialized settings. Zether [25] implements a variant mechanism of Bulletproof to hide the transaction amount and addresses with the help of EIGamal encryption. But the computational and verification costs of its provers are still much higher than those of zk-SNARKs. We found that Groth16 [26] can generate proofs with low time complexity and space complexity. Therefore, we utilize the Groth16 proof system to achieve zero-knowledge proofs [27].

Federated Learning provides a possibility to construct a global model while preserving the training data. The literature [27], [28], [29] propose the distributed frameworks by combining blockchain with federated learning while maintaining the security and trustworthiness of blockchain. At the same time, the communication efficiency in the blockchain is also optimized. Some scholars [28] propose a secure architecture authorized by blockchain for sharing data, which stores the federated models through blockchain and ensures the security of the data sharing process. The article [29] proposes a new privacy-preserving mechanism and designs a two-stage solution that includes the transformation of intelligent data and detection of collaborative data leakage The work is proven to be high accuracy, efficiency, and reliability. Inspired by the above works, we propose the proof of quality, i.e., POQ, to improve the efficiency of federated training.

3 Multi-level blockchain system privacy protection mechanism

3.1 Privacy protection based on zero-knowledge proof and double balance

The architecture of this system includes three types of nodes: regulators, commercial banks, and suppliers who need the loan. Two types of blockchains are composed from those nodes. Different types of nodes publish different content by their roles. The constructed architecture of the system is shown in Figure 1:

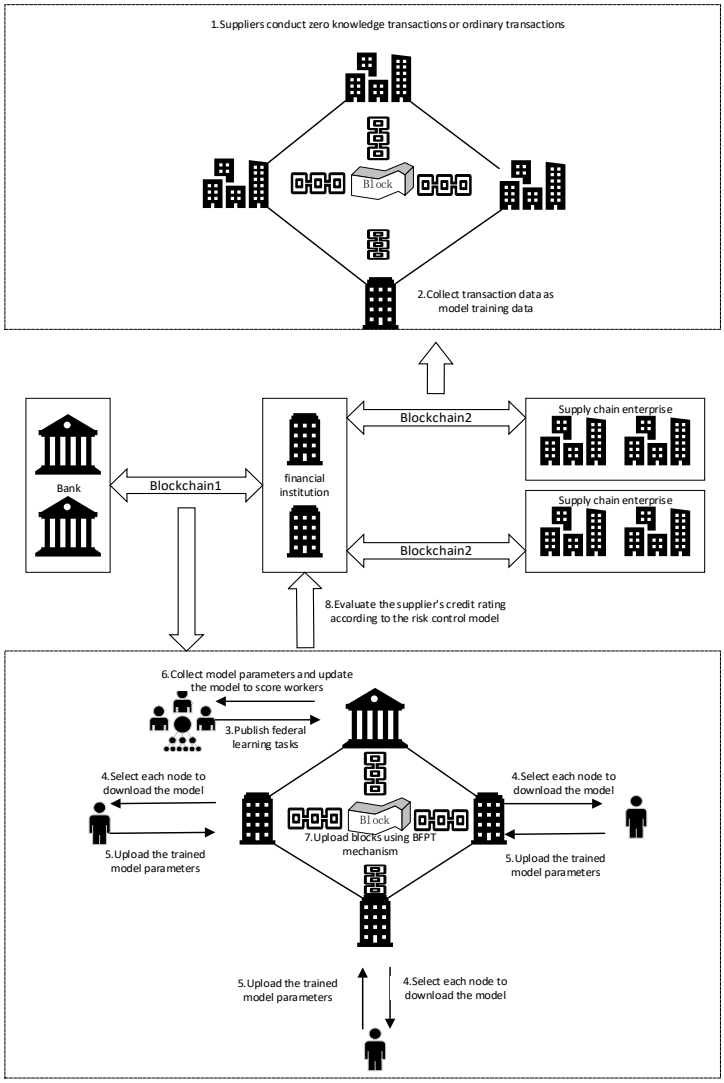

Figure 1 Illustration of Transaction system

Blockchain 1 is the blockchain between commercial bank and lender supplier: it consists of commercial banks and lender suppliers. And it is used to maintain the transaction records among enterprises. In addition, it can hide important transaction information through zero-knowledge proof and protect supplier privacy by setting up a double-balanced transaction model. Meanwhile, reputation rating is introduced. The suppliers with different reputation scores have different rights when posting transactions. Suppliers with low reputation scores will be faced to cancel the permission of zero-knowledge proof. Vendors with a very poor reputation will have their transaction privileges removed.

Blockchain 2 is the blockchain between commercial bank and regulator: its function is mainly to give a reputation score to each lender. This chain combines the features of federal learning and blockchain. The regulator acts as a task publisher and the commercial banks are viewed as participants in Federal Learning. The publisher first sends the initial global risk control model to the participants in the form of outsourcing. Then, during per round of collaborative training, each participant iteratively trains his local model without uploading his private training data to the publisher. In contrast, the parameters of the local model are sent to the publisher. Then the publisher aggregates the parameters from all participants to update the global model. After that, the publisher uploads the global model to Blockchain 2 forguaranteeing the global model to be immutable. We also introduce a quality metric to select participants according to the quality scores by POQ That can avoid some malicious nodes affecting the performance of the overall model. All commercial banks form a consensus committee, which uses the PBFT consensus algorithm to verify the blocks.

3.2 Privacy protection based on zero knowledge proof and double balance

This section introduced in detail how to protect transaction privacy of the transaction system in blockchain 1 based on ZKP and double-balance technology. Common symbols used in the paper are shown in Table 1 below.

Table 1 symbol definition

Expression	Description
A	Sender Alice
B	Recipient Bob
sk_A	The private key of A
pk_A	The public key of A
$addr_A$	The address of A
$pt\_balance_A$	The plaintext balance of A
$zk\_balance_A$	Zero knowledge balance of A
$Ledger_T$	Account book
TDSet	Global data table
SNSet	Serial number table
CRH	Anti-collision hash function
$COMM_{bc}$	A non-interactive commitment scheme for hiding account balances

$COMM_{tc}$	A non-interactive commitment scheme for hiding transfer amounts
cmt_A	Express balance commitment of A
$value_A$	The plaintext balance of A
sn_A	Serial number associated with cmt_A
r_A	Random number confusing sn_A
PRF	Pseudorandom function
cmt_v	Commitment to transfer the amount

Entities are defined as follows:

- Ledger

Anyone can access Ledger at any time. It is a block containing all transactions

- Address key pair

Each user has a pair of address keys (*sk, pk*). Account address:addr: = *CRH (pk)*

- Commitment

Balance commitment refers to the commitment to account balance, i.e. $cmt_A = COMM_{bc}(addr_A, value_A, sn_A, r_A)$. $COMM_{bc}$ is a non interactive commitment scheme for statistical hiding of account balance, with hidden and bound attributes.

.

- Public data Table

The public data table is used to store one-time variables including *TDSet* and *SNSet.* $TCMSet_N$ represents the data sheet of all publicly announced fund transfer commitments cmt_v in block $block_N$.

Definition of zk-SNARK:

The zk-SNARKs scheme [12]-[14] can be represented as the following polynomial tuple:

$$\Pi_Z(\text{Setup}, \text{KeyGen}, \text{GenProof}, \text{VerProof})$$

The transaction algorithm are as follows:

- Setup $(1^\lambda) \rightarrow pp_Z$: Preset a random security parameter λ, the algorithm will generate and output a public parameter list $pp_Z = (\mathbb{F}_p, p, \mathbb{G}_1, \mathcal{P}_1, \mathbb{G}_2, e, \mathcal{P}_2, \mathbb{G}_T)$, where $\mathbb{F}_P$ is a finite field and p is a prime number; $(\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T)$ are cyclic groups of order p; $\mathbb{G}_T$ is derived from $\mathbb{G}_1 \times \mathbb{G}_2$; e is a bilinear pair; $\mathcal{P}_1$ and $\mathcal{P}_2$ are generators of $\mathbb{G}_1$ and $\mathbb{G}_2$ respectively; $\mathbb{F}_P$ is a finite field.
- Keygen (c) $\rightarrow (pk_Z, vk_Z)$: Preset a circuit C, and the algorithm generates a key pair (pk_Z, vk_Z) via the public parameter pp_Z as theproof key and verification key of the zero-knowledge proof.
- Genproof $(pk_z, \vec{x}, \vec{a}) \rightarrow \pi$: Preset key pk_Z, a state declaration $\vec{x}$ and a private evidence $\vec{a}$, this algorithm is used to generate a zero knowledge proof π, and returns $\perp$ if it fails. Among the input parameters, pk_z is the proof key to generating the zero-

knowledge proof; $\vec{x}$ and $\vec{a}$ are used as the inputs of circuit C; π is the zero-knowledge proof, which can prove that $\vec{x}$ and $\vec{a}$ indeed meet the relationship constructed by circuit C.

- Verproof $(vk_Z, \vec{x}, \pi) \rightarrow$ B: Preset key vk_Z, the state declaration $\vec{x}$ and zero knowledge proof π, this algorithm can verify the correctness of zero knowledge proof. If zero knowledge verification is successful, output B = 1; Otherwise, output B = 0.

The Main functions are as follows:

- Setup(1^λ)→ pp;Preset the security parameter λ, the algorithm generates the system public parameter pp with number of λ bits by a trusted third party, and the parameter pp is public.
- CreateAccount(pp)→$\{addr,(sk,pk),tt\}$; Preset the public parameter pp, the algorithm creates an account address $addr$ for the user and generates a key pair (sk, pk), in which the private key sk is used to access private data and decrypt the ciphertext data in the transaction,and the public key pk is used to encrypt the transaction data to be submitted, and tt is adopted as a traceable label,which is shared with commercial banks and regulatory authorities, and enables them to track the transaction.
- Mint (pp, $zk\_balance_A, sk_A, v$) → $\{zk\_balance_A^*, tx_{Mint}\}$; This algorithm enables account A to convert plaintext amount v into-zero knowledge amount and merge it with the current zero-knowledge balance.
- Redeem($pp,zk\_\ balance_A, sk_A, v\ ) \rightarrow \{zk\_balance_A^*, tx_{Redeem}\}$; This algorithm enables account A to send zero-knowledge balance into plaintext balance and merge it with the current plaintext balance. The operation is similar to Mint.
- Send($pp,zk\_balance_A, sk_A, pk_B, v, tt$)→ $\{zk\_balance_A^*, tx_{Send}\}$; This algorithm enables sender a to send zero-knowledge amount to receiver B. After $\mathrm{tx_{Send}}$ transaction is generated by hiding the transaction amount and receiver address, account A informs account B offline of the transaction hash value $h_{tx_{send}} := CRH(\mathrm{tx_{Send}})$, so that account B can retrieve and parse $\mathrm{tx_{Send}}$. At the same time, the regulator can track the receiver through the tracking tag ta_B. The process is shown in Figure 2:

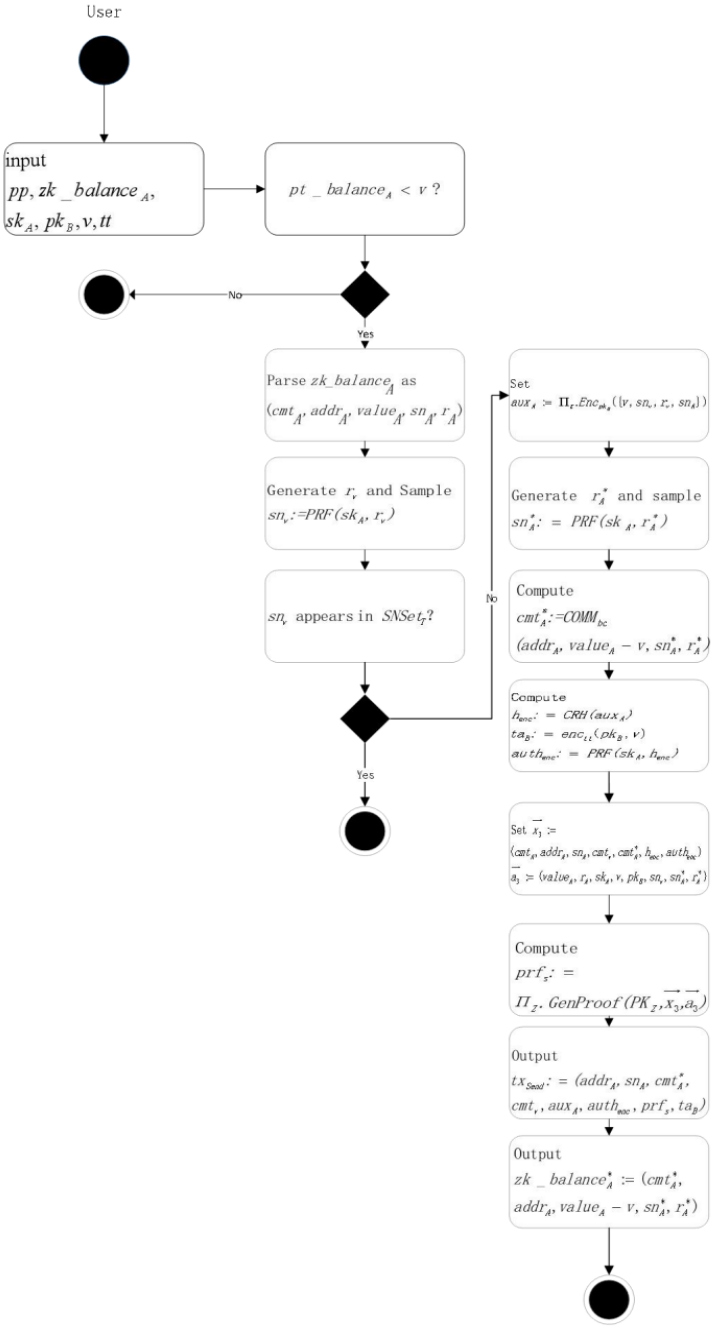

Figure 2 Send

- Deposit ($Ledger_T$, pp, (sk_B, pk_B) , $h_{tx_{Send}}, zk\_balance_B, tt) \rightarrow \{zk\_balance_B^*, tx_{Deposit}\}$; This algorithm allows recipient B to check the collection and deposit the received amount into its own account. Preset the current Ledger, public parameter pp, the account key pair (sk_B, pk_B), the hash value $h_{tx_{send}}$ of tx_{Send} and current zero-knowledge balance of account B $zk\_balance_B$, It is possible for receiver B to call the Deposit algorithm to make a collection deposit to obtain a new zero-knowledge balance $\text{zk\_balance}_{\text{B}}^*$ and generate the transaction $\text{tx}_{\text{Deposit}}$, According to the transaction hash value $h_{tx_{send}}$ generated by sender A, receiver B is allowed to retrieve and parse tx_{Send} to construct $\text{tx}_{\text{Deposit}}$ for deposit. The operation is similar to Send.
- Vertx $(Ledger_T, pp, tx) \rightarrow$ b; Preset the current ledger, public parameter pp, transaction tx. It is possible for verifier to use this algorithm to check whether all zero-knowledge transactions are valid. If valid, output 1, invalid output 0.

3.3 Construction of federal learning risk control model

Federated learning can ensure that the training data of participants are not out of the local while training the global model, so as to protect the data privacy effectively. The federated learning system architecture in this thesis includes task publishers, task participants and blockchain 2. Publishers and participants need to register accounts on the blockchain to obtain their own public-private key pairs.

Definition of quality score：

Because the scheme proposed in this thesis is open, an effective management mechanism is needed to evaluate the quality of federated learning between nodes, so as to eliminate the situation that malicious nodes may destroy the whole federated learning task. Here a quality scoring algorithm POQ is established to score each node.

In this quality scoring algorithm POQ, the task publisher scores the participants according to the impact of the participants on the model quality in each task. Among them, the model uploaded by participants after training through local data which improves the accuracy of the overall model, is recorded as a positive interaction event, on the contrary, the model which reduces the accuracy of the model is recorded as a negative interaction event. α_i represents the number of positive interaction events made by participant *i*, and the initial value is 1. β_i represents the number of negative interaction events made by participant *i*, and the initial value is 0. b_i indicates the credibility of participant I. Because both positive interaction events and negative interaction events will occur in federal learning tasks. However, from the actual results, negative interaction events can usually lead to serious consequences. Therefore, negative interaction events have high weights. So let K and t be the weights of positive interaction events and negative interaction events respectively, where k = 2 and T = 3, so

$$b_i = \frac{ka_i}{ka_i + t\beta_i} \tag{1}$$

Suppose the initial score of each node as p = 20 and the reliability weight as μ = 50. Considering that some nodes cannot participate in the following training due to the

vicious circle caused by the poor training effect of the model at the beginning, a compensation score is added to each node (suppose s as the training number, y_i as the last training number of participant i, the weight h = 0.8, and the compensation is ((s-y_i) * h) ≤ 30), so as to ensure that each node has the opportunity to participate in training. Therefore, the quality score S_i of participant i by the s time training is:

$$S_i = p + \left(s - y_i * h + \frac{\mu k a_i}{k a_i + t\beta_i}\right) \tag{2}$$

The federal learning algorithm is as follows:

Algorithm 1 federated learning algorithm

input:

The current quality score of all nodes ($S_1, S_2, S_3, .. S_n$), n is the total number of all nodes in blockchain 1.

A set of participants w requesting participation in federal learning;

Global iteration times of Federated learning task, N;

output:

Final accuracy of federal learning tasks, a;

Quality score of all participants ($S_1, S_2, S_3, .. S_n$);

1: The federal learning task publisher publishes the task and receives the W;

2: The task publisher obtains the quality score of each node in W;

3: The task publisher selects K nodes with the highest scores as participants according to the quality score S_i of all participants in W or special requirements to obtain the final participant set n

4: for $i = 0; i < k; i++$ do

5: The task publisher publishes the global model to the participants in the;

6: Participants in set n download the model and train the model through local data.

7: After training, upload the trained model;

8: The task publisher updates the global model from the collected model data;

9 end for

10: After this task, update the quality scores of all nodes ($S_1, S_2, S_3, .. S_n$) according to the POQ algorithm;

11: return A and ($S_1, S_2, S_3, .. S_n$);

Steps 1 to 3: Release federated learning tasks. The task publisher broadcasts the federated learning tasks it needs and specific data requirements. After receiving the broadcast, each participant node decides whether to apply to the federal learning task. Then the publisher calculates the quality score of the participant node requested to apply, and selects an appropriate number of participants to participate in federal learning and training according to the quality score or special requirements.

Steps 4 to 10: Conduct federated learning. The task publisher publishes an initial machine learning model as a global shared model. After each participant trains his own local data and uploads new model parameters, the task publisher tests the accuracy of the model through the test set and updates its quality score.

Step 11: Update the final model and upload the model and training records to the blockchain. The publisher submits the legal but unconfirmed relevant data generated in this round of asynchronous global model training to the block. The publisher digitally signs and broadcasts the packaged blocks and enters the block verification stage. Finally, the consensus committee verifies the effectiveness of the blocks through the PBFT consensus mechanism.

4 Experimental design and evaluation

Construction of accounts receivable financing scheme:

Accounts receivable financing is a type of financing arrangement in which suppliers who need funds (i.e. financiers and general suppliers) conditionally transfer or pledge accounts receivable, and commercial banks and other lending institutions provide financial services such as financing funds to enterprises. Here, we set the following 8 nodes:

The blockchain network consists of four users and four administrators. The accounts receivable financing process is as follows: first, the upstream suppliers in the supply chain sign a purchase agreement with the core enterprise, stipulating that the upstream suppliers provide with goods, and the core enterprise pays within a certain time after receiving the goods. Every transaction between suppliers and the core enterprise goes up the chain. Secondly, upstream suppliers, as financiers, apply for loans from commercial banks and other financial institutions, and then the core enterprises make repayment / repurchase commitments. Banks and credit rating agencies can check the capital transactions between suppliers and the core enterprise, and finally commercial banks decide whether to lend loans according to the evaluation results.

We build an 8-node multi-layer blockchain through FISCO alliance chain [30], and each organization has two nodes. Node A is the commercial bank node. Node B is the regulatory authority node. Node C and node D are upstream supplier nodes. Blockchain 1 establishes a risk control model through federated learning in the form of smart contract, and scores nodes C and D through the risk control model. This thesis sets up three virtual machines (Ubuntu 20.04LTS system AMD Ryzen 5 2600x, 32gNvidia1650SUPER), and carries out cooperative training of federal learning through the federal learning platform FATE [31].

Federated learning experiment design：

Blockchain 1 is used in the federal learning experiment, and the horizontal federal learning experiment is carried out by using the built-in online loan data set of FATE federal learning platform. The logistic regression model is used as the training model. Set 1 task publisher and 2 task participants. Figure 3 shows the results:

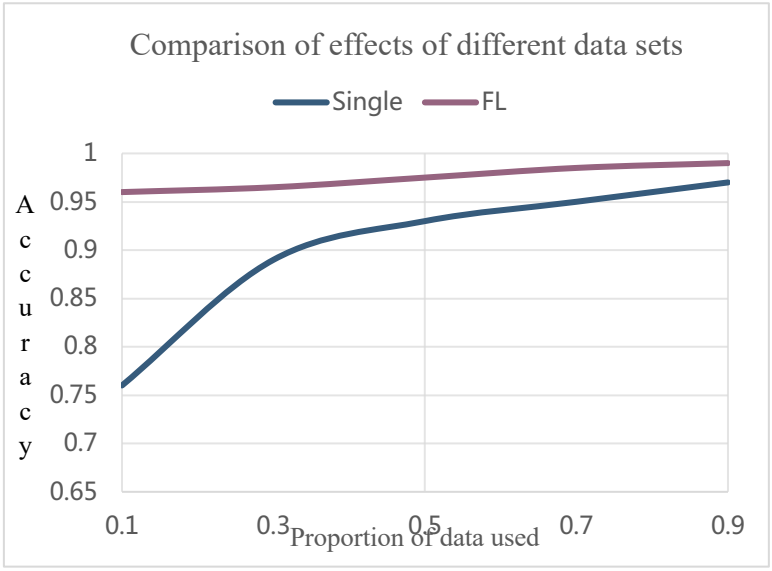

Figure 3 Federal Learning results

Zero-knowledge proof experiment design:

Organization C and organization D are supplier nodes. Here, they use the double balance transaction model based on zero-knowledge proof to hide the transaction amount and the other party's address, so as to achieve privacy protection. At the same time, commercial bank node A has the right to cancel the zero-knowledge proof transaction permission of C or D. Here, we use Libsnark cryptography library to implement zk-SNARK scheme. zk-SNARK are used to generate zero-knowledge proofs for the circuits constructed by these transactions. The key pair used for zk-SNARK generation or verification proof is preinstalled on each node.

For example, supplier A and supplier B register accounts and obtain account addresses and key pairs respectively. When A wants to initiate an account transfer to B, A transfers the plaintext balance X to zero-knowledge balance by generating a balance commitment and initiates a Mint transaction. Each node links the transaction after verifying that the commitment is valid through the BPFT consensus mechanism. Then A generates a new zero-knowledge balance commitment and transfers the zero-knowledge amount v to B through Send. The transaction is linked, and A informs B of the hash value of the transaction offline. Therefore, node B can analyze the transaction with its own key and transfer the zero-knowledge amount transferred by A to its own zero-knowledge balance.

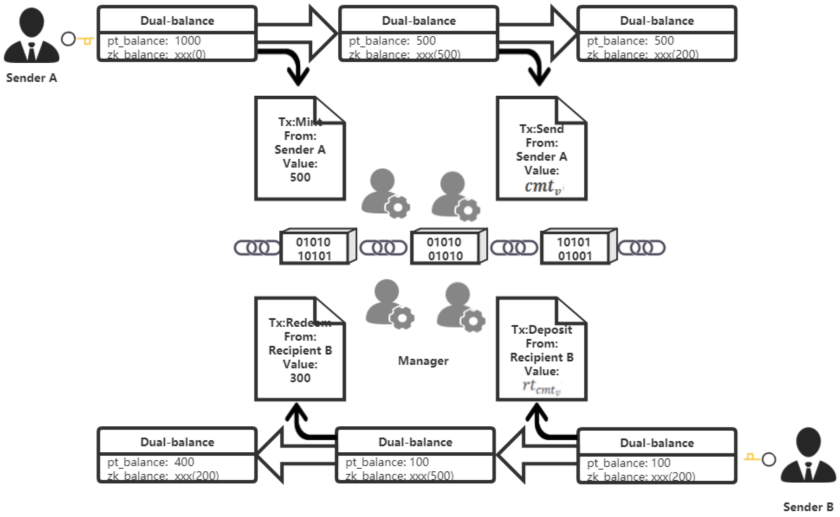

Figure 4 Transaction workflow

Blockchain 2 is used in the transaction experiment. We compared the zero-knowledge proof application Zerocash on the market. The test is conducted from the two dimensions of time and space, and the results are shown in Table 2. The above experiment shows that our privacy protection mechanism can effectively improve the operation efficiency of most systems while ensuring account privacy, and federated learning can improve the model training efficiency while protecting data privacy.

Table 2 Comparison Experiment of zero knowledge proof transaction

Zero-knowledge proof system			Zerocash system		
Setup	Time	97.4s	274.3s	Time	**Setup**
	pp	266MB	1.87G	pp	
Create Account	Time	1ms	743ms	Time	**Create create**
	Pk	64B	343B	Addr	
Mint	Time	4.82s	1μs	Time	**Mint**
	Tx	357B	72B	tx	
Redeem	Time	4.73s	104.2s	Time	Pour
	Tx	357B			
Send	Time	8.51s	1004B	tx	
	Tx	509B			
Deposit	Time	18.75s	2.14ms	Time	Receive
	Tx	433B			
Vertx	Time	14.52ms	28.63ms	Time	**Verify**

5 Conclusion

In the financial transaction system, a single blockchain system cannot meet the actual needs. Therefore, this thesis constructs a multi-level blockchain system to meet the application needs. Aiming at the privacy protection problems involved in blockchain transactions, a double balance transaction model based on zero knowledge is adopted to ensure the transaction privacy between suppliers. In addition, federal learning is used to solve the problem of "data island" between financial institutions and protect data privacy. Finally, through theoretical analysis and a series of comparative experiments, the effectiveness of the scheme is verified and the potential applications are fantastic.

6 References

1. Nakamoto S . Bitcoin: A Peer-to-Peer Electronic Cash System. 2009.
2. Chao L , He D , Huang X , et al. BSeIn: A blockchain-based secure mutual authentication with fine-grained access control system for industry 4.0[J]. Journal of Network and Computer Applications, 2018, 116:42-52.
3. Wang H , Song X , Junming K E , et al. Blockchain and Privacy Preserving Mechanisms in Cryptocurrency[J]. Netinfo Security, 2017.
4. Zhang Y , Lu Y , X Huang, et al. Blockchain Empowered Asynchronous Federated Learning for Secure Data Sharing in Internet of Vehicles[J]. IEEE Transactions on Vehicular Technology, 2020, PP(99):1-1..
5. JK You, Hong C S . Blockchain-based Node-aware Dynamic Weighting Methods for Improving Federated Learning Performance[C]// 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE, 2019.
6. EB Sasson, Chiesa A , Garman C , et al. Zerocash: Decentralized Anonymous Payments from Bitcoin[C]// 2014 IEEE Symposium on Security and Privacy (SP). IEEE, 2014.
7. Nicolas Saberhagen N.Cryptonote v2.0[EB/OL], https://cryptonote.org/whitepaper.pdf,2013.
8. Miers I , Garman C , Green M , et al. Zerocoin: Anonymous Distributed E-Cash from Bitcoin[C]// Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013.
9. Duffield E, Diaz D. Dash: A Payments-Focused Cryptocurrency[EB/OL], https://github.com/dashpay/dash/wiki/Whitepaper, 2016.
10. Maxwell G. CoinJoin: Bitcoin privacy for the real world[EB/OL], https://bitcointalk.org/index.php?Topic=279249.0, 2013.
11. BitInfoCharts. Cryptocurrency statistics[EB/OL], https//bitinfocharts.com/.
12. Parno, Bryan, AUTHOR, et al. Pinocchio: Nearly Practical Verifiable Computation[J]. Communications of the ACM, 2016, 59(2):103-112.
13. Groth J , Maller M . Snarky Signatures: Minimal Signatures of Knowledge from Simulation-Extractable SNARKs[C]// International Cryptology Conference. 2017.
14. Miers I , Garman C , Green M , et al. Zerocoin: Anonymous Distributed E-Cash from Bitcoin[C]// Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013. Buterin V. Ethereum white paper: a next generation smart contract & decentralized application platform[J].First version, 2014.
15. Buterin V. Ethereum white paper: a next generation smart contract & decentralized application platform[J].First version, 2014

16. Fleder M , Kester M S , Pillai S . Bitcoin Transaction Graph Analysis[J]. Computer Science, 2015.
17. Chiesa A , D Ojha, Spooner N . Fractal: Post-quantum and Transparent Recursive Proofs from Holography[M]. Springer, Cham, 2020.
18. Boew S, Grigg J, Hopwood D. Halo: Recursive Proof Composition without a Trusted Setup[J].IACR Crptology ePrint Archieve, 2019.
19. B Bünz, Fisch B , Szepieniec A . Transparent SNARKs from DARK Compilers[C]// 2020.
20. Maller M , Bowe S , Kohlweiss M , et al. Sonic: Zero-Knowledge SNARKs from Linear-Size Universal and Updatable Structured Reference Strings[C]// the 2019 ACM SIGSAC Conference. ACM, 2019.
21. Chiesa A , Hu Y , Maller M , et al. Marlin: Preprocessing zkSNARKs with Universal and Updatable SRS[C]// Annual International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Cham, 2020.
22. Gabizon A, Williamson Z J, Ciobotaru O.PLONK: Permutations over LAgrange-bases for Oecumenical Noninteractive arguments of Knowledge[J].IACR Cryptology ePrint Archive, 2019.
23. Bunz B , Bootle J , Boneh D , et al. Bulletproofs: Short Proofs for Confidential Transactions and More[C]// 2018:315-334.
24. B Bünz, Agrawal S , Zamani M , et al. Zether: Towards Privacy in a Smart Contract World[C]// 2020.
25. Groth J , Maller M . Snarky Signatures: Minimal Signatures of Knowledge from Simulation-Extractable SNARKs[C]// International Cryptology Conference. 2017.
26. G Zhangshuang. Research on privacy protection of account model blockchain system based on zero knowledge proof[D]. ShanDong: Shandong University,2020.
27. Lu Y , Huang X , Dai Y , et al. Blockchain and Federated Learning for Privacy-Preserved Data Sharing in Industrial IoT[J]. IEEE Transactions on Industrial Informatics, 2019, PP(99):1-1.
28. JK You, Hong C S . Blockchain-based Node-aware Dynamic Weighting Methods for Improving Federated Learning Performance[C]// 2019 20th Asia-Pacific Network Operations and Management Symposium (APNOMS). IEEE, 2019.
29. Lu Y , Huang X , Dai Y , et al. Federated Learning for Data Privacy Preservation in Vehicular Cyber-Physical Systems[J]. IEEE Network, 2020, 34(3):50-56.
30. Webank-fisco-bcos.Cryptocur. An alliance blockchain underlying technology platform[EB/OL], https://github.com/FISCO-BCOS/FISCO-BCOS.
31. Webank-fate. An open source federated learning platform[EB/OL], https://github.com/FederatedAI/FATE.