EasyChair Preprint

№ 3086

A Case Study for Blockchain in Contract: FutureText Prototype for Electronic Contracts Signing and Data Recording

Qing Zhang, Jian Gao, Qiqiang Qin, Chenyu Wang and Keting Yin

March 31, 2020

A Case Study for Blockchain in Contract: FutureText prototype for electronic contracts signing and data recording

Qing Zhang[1,2], Jian Gao[1,2], Qiqiang Qin[1,2], Chenyu Wang[1,2], Keting Yin[3*]
[1] China Financial Futures Exchange
[2] Shanghai Financial Futures Information Technology Co., Ltd
[3] College of Software Technology, Zhejiang University
zhangqing@cffex.com.cn, gaojian@cffex.com.cn, qinqq@cffex.com.cn, wangcy1@cffex.com.cn, yinkt@zju.edu.cn

Abstract

This paper proposes a convenient, economical and environmentally friendly way to complete the Privately Offered Fund contracts based on shared permission blockchain. There are three main contributions of applying blockchain technology into the contracts. First, in order to reduce the CO_2 emission from printing, delivery and mostly for the sake of security, we present an electronic procedure for signing and keeping the Privately Offered Fund Contracts. Second, we design and implement FutureText prototype using Hyperchain, which helps us share privacy information among stakeholders while keeping forged signature, counterfeited seal and twin-contracts away. Finally, to follow the legitimacy and compliance, we put forward methods for both authentication and signature of tripartite contracts. Besides, we put encrypted contracts on blockchain while keeping original contracts on local hosts using heterogeneous storage, and that helps to alleviate the storage pressure on blockchain.

Key words-blockchain, electronic contract, authentication, tripartite signature, heterogeneous storage

1 Introduction

With China's rapid economic development and deepening of reform and opening-up, the Privately Offered Fund has grown into a huge market: by November 2018, the number of registered private fund managers exceeded 24,418 with the number of registered private equity funds reached 75,220;

* Corresponding author: Keting Yin; Email:yinkt@zju.edu.cn

meanwhile the market scale was 12.79 trillion Chinese Yuan[†]. However, with the rising of the market, the paper print contracts are revealing more and more defects, such like the huge paper consumption and costly delivery as well as the risk in missing contracts and counterfeit contracts. Considering the scale and complexity of the business, traditional paper print contracts become unsuitable and unmanageable for the business.

Our paper aims to solving the problems by introducing an electronic signing and data recording system named FutureText based on a shared permission blockchain.

Overall, the main contributions of the paper can be summarized as follows:

-First, we introduce blockchain into the Privately Offered Fund business, making business process convenient and contract signing trustable.

-Second, we put up an authentication way to identify the contractors for compliance with relevant laws and regulations.

-Last but not the least, we propose a method to sign tripartite contracts. Through this method, our system can protect contractors' privacy while achieving consensus.

The rest of paper is divided into following parts. In section 2, we define the problem of the Privately Offered Fund business and several key concepts. We will illustrate the advantages of using blockchain in contracts. In Section 3, related work on electronic contract is introduced. Most work concentrates on model building and legal issues. In Section 4, we give an overview of our system design. In Section 5, the specific program about FutureText is given. More details on FutureText is discussed such as how to solve the authentication problem and how to complete a tripartite contract's consensus. In Section 6, the performance and latency of FutureText are presented. Finally, we conclude with discussions in section 7.

2 Motivation

The Privately Offered Fund (POF) is a securities investment fund that raises money from specific investors in a non-public manner and bases on specific underlying assets. In recent years, POF market in China has been growing rapidly. Typically, managers, custodians and investors ensure their respective rights and obligations by signing a POF contract. While, as the first step of the whole process, getting real market quotes and seeking the counterparty become a time consuming task since POF works in a non-public manner and information asymmetry for specific underlying assets.

Even though we get a reasonable quote and reliable counterparty, there is still hundreds of pages legal terms and conditions to read and modify among managers, custodians and investors. Under this circumstance, it is inconvenient and hard to manage the life cycle of a tripartite contract. Here is a case: A manager is in charge of drafting, printing and delivering the contract. The investor is probably on vocation or a business trip. The custody might find defaced contracts or faked signature and ask for re-signing. All reasons above could lead to paper consumption and time out in contract signing.

According to regulations, filing of POF in China Securities Investment Fund Association is necessary. The regulatory authorities have privileges on tracking business operations. Archived contracts will benefit in conducting pre-product, in-product, and ex-product supervision. The regulatory authorities can figure out problems such as fraud, insider trading and illegal agency holding in the first place.

In the end, after the signatures of contracts among managers, custodians and investors, the contracts are required to preserve for another decade after they are expired. Obviously, the long range of preserving need large resource consumption, including the space and people for caring without considering the paper aging, loss and leaking. Besides, if needed, searching among bunches of papers is inefficient in most cases.

[†] Data comes from China Securities Investment Fund Association

To solve the problems above, we put forward a shared permission blockchain based electronic contract signing and data recording system named FutureText. FutureText will help us in the following aspects.

Firstly, FutureText supports online signing of electronic contracts, which means the reduction of the cost in printing, delivering and preserving. Besides, FutureText enhances the trust and safety on tripartite contracts.

Secondly, with decentralization, non-tamperability and traceability of blockchain, FutureText is good at information sharing and procedure tracing.

Finally, FutureText is access to the regulatory authorities for supervision.

3 Related Work

In recent years, many scholars have researched the application of electronic contracts from the perspective of both law and technology. Y. K. Zhou[1] demonstrates the legitimacy of electronic contracts in the form of data, and proposes that the validity of electronic contracts should be based on functions, not the uniformity of forms. T. Ge[2] denoted the necessary conditions for an electronic contract to be legally valid evidence. It should satisfy the pre-preparation, the clear record of the matter, and it is able to trace the whole process of signing the contract afterwards. X. M. Zhen[3] discussed the legal issues faced by electronic contracts, from the withdrawal and revocation of electronic commitments, electronic signatures and the use of electronic contracts as evidence.

Besides the law and technology, many scholars also studied the application of cryptography to achieve electronic contract signing. R. Xu, et al[4] designed an electronic contract service platform based on tamper-resistant technology. The platform ensured the uniqueness of the user identity, the integrity of the transmitted data, and the immutability of signatures on contracts by combining user identity authentication, electronic contract encryption transmission, and notarization participation. J. L. Cai, et al[5] created a multi-party cross-regional and all-weather contract signing platform. The platform combined functions including identification, digital certificate management, electronic signing of contracts, and electronic deposit of contracts. Z. C. Yin, et al[6] explored the feasibility of using blockchain technology in electronic contract signing, and implemented the electronic contract signing system based on the Hyperledger blockchain platform. J. T. Gu, et al[7] analyzed the development of e-commerce in China, as well as the current status of e-contract, electronic signature and online negotiation in China. J. T. Gu, et al[7] also proposed an electronic contract signing system based on electronic signature to realize the electronic life cycle of the signing process, including signing, saving and verification. A. Burunova et al[8] compared four different electronic contract models. Different models were designed for different application scenarios

Along this line, this paper will continue to study electronic contracts signing and data recording and propose another solution to the issues for the Privately Offered Fund business.

4 System Design Overview

4.1 System Architecture

The system architecture is divided into four layers as shown in Figure 1:

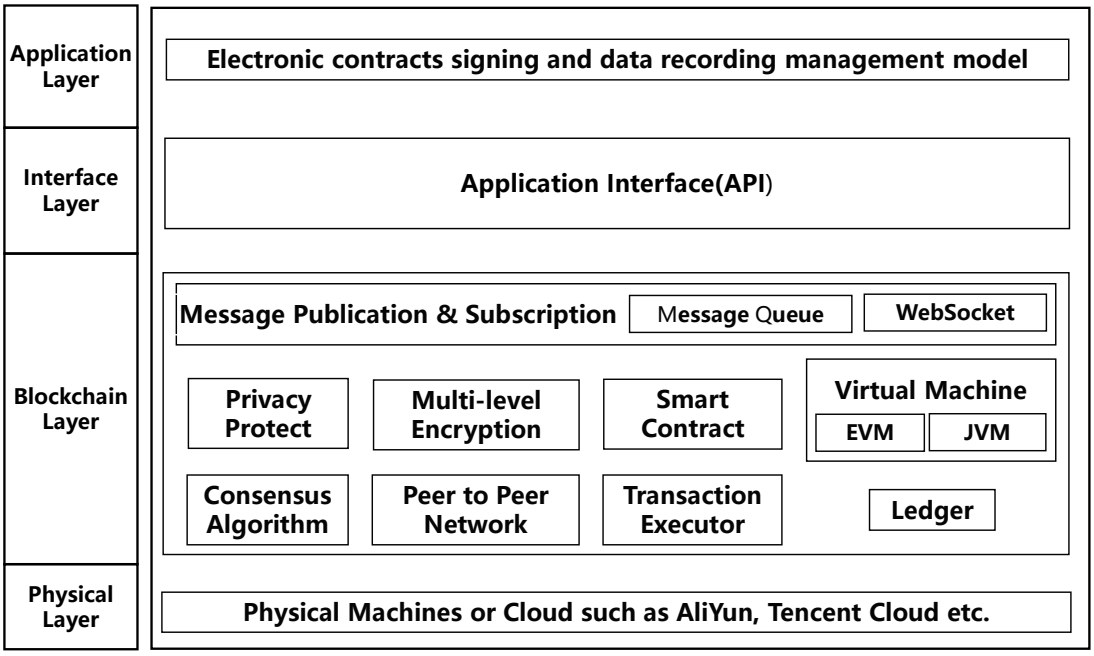

Figure 1: System Architecture of FutureText

1) *First Layer: Physical Layer.* The first layer is the Physical Layer, which consists of physical machines at local and virtual machines offered by Cloud Service Providers. The system uses a hybrid deployment mode combining physical machines and cloud hosts. In order to satisfy byzantine fault tolerance, the system deploys at least four blockchain nodes. In addition, an extra machine is needed to provide external access and deploy web applications.

2) *Second Layer: Blockchain Layer.* The second layer is Blockchain Layer, which means blockchain as a service (BaaS). The blockchain platform support virtual machines such as EVM (Ethereum Virtual Machine) and JVM (Java Virtual Machine). The Smart Contract and Transaction Executor support solidity language and Java language. The key information, such as contract hash, signing body is shared on blockchain with privacy protecting among related parties. We can interact with Smart Contract via Java API. Besides, message publication and subscription is essential for communication with outsiders.

3) *Third Layer: Interface Layer.* The third layer is Interface Layer, which mainly provides a way to interact with the application. FutureText can call methods in Smart Contract through Java program. Besides, if the blockchain platform provides a variety of interfaces for querying blockchain data, this layer can be used to obtain data stored on blockchain through block hash or transaction hash.

4) *Fourth Layer: Application Layer.* The fourth layer is Application Layer, which is the core part of the FutureText. The Application Layer mainly consists of electronic contracts signing and data recording management model. We will discuss that model later in this section.

4.2 Shared Permission Blockchain: Hyperchain

Our work is based on the shared permission blockchain platform: Hyperchain. Hyperchain is a leading enterprise-level blockchain platform developed by Hangzhou Qulian Technology. As shown in Figure 1, Hyperchain provides a package of solutions: Privacy Protect and Multi-level Encryption help improve safety; Message Publication and Subscription guarantee the interactivity; Smart Contract and

Virtual Machines allow users to update the Ledger and write details onto blockchain. The Consensus Algorithm is robust byzantine fault tolerance (RBFT). RBFT makes sure nodes in blockchain network process transactions submitted from clients in the same sequence. The consensus process of RBFT is shown in Figure 2.

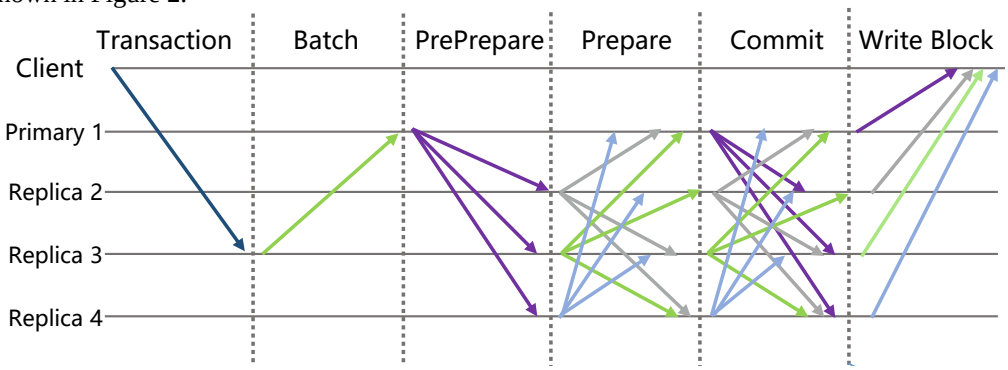

Figure 2: Procedure of RBFT Consensus Algorithm

Suppose there are 3f+1 nodes in blockchain network, where f is the node number. RBFT can tolerate errors of f nodes. In Figure 2, Primary node is selected dynamically. Primary node manages the package of clients' transactions in sequence. Replica nodes are only for backup, which consist with Primary node in transaction. Once the Primary node is out of service, one of the Replica nodes will automatically become the new Primary node.

As the case showed in Figure 2, RBFT procedure has six stages. In Transaction stage, Client sends transaction via its connected node. In Batch stage, Replica 3 receives the transaction and broadcasts the transaction to Primary 1 instead of all nodes. Primary 1 receives the transaction and verifies it. Once verified, Primary 1 batch processes all transactions and package them in sequence. In PrePrepare Stage, Primary 1 broadcasts the packaged transactions to all nodes. In Prepare Stage, all Replicas receive the packaged transactions and send Prepare message to other nodes, if Replica agrees with the batching and ordering result of Primary 1. In Commit stage, if Replica receives 2f Prepare messages and verifies the legality from Primary 1 the Replica will send Commit messages to other nodes. In Write Back stage, if Replica receives 2f+1 Commit messages, it means all nodes reach a consensus. Replicas will call transaction executor and compare it with the result from Primary. Once verified, the results are written into the ledger.

The benefits of adopting RBFT has two main aspects. First, when receiving a new transaction, Replica forwards it to Primary instead of the entire network and that reduces the bandwidth consumption. Second, the illegal transactions will be removed if the Primary finishes validation. It makes sure that illegal transactions do not consume the blockchain network computing power. Therefore, by applying RBFT, Hyperchain can effectively alleviate the burden of the whole network.

According to practice, Hyperchain has a high-performance of at most 10000 transactions per second and its best transaction execution time is around 300 milliseconds.

4.3 Electronic Contracts Signing And Data Recording Model

The electronic contracts signing and data recording model used in FutureText consists of four modules as shown in Figure 3.

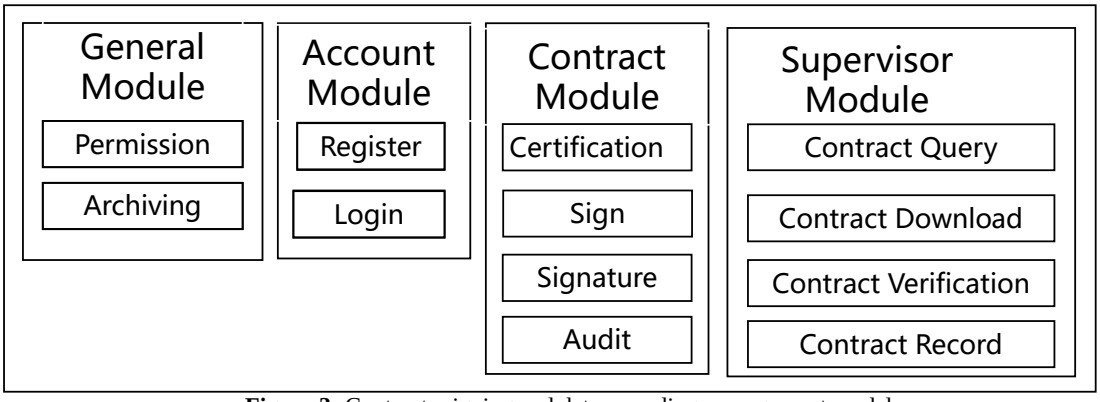

Figure 3: Contracts signing and data recording management model

(1)*The General Module*. In this module, Permission is responsible for controlling who has the right to see full details of the contract. In FutureText, only the involved parties and the supervisory authority have access to privacy details. Archiving step will package the encrypted contracts and remove it away from current blockchain.
(2)*The Account Module*. In this module, we have Register and Login service. Register allows personal users and agencies to submit identification information. Login provides access to the certificated users and agencies.
(3)*The Contract Module*. In this module, we have Certification, Sign, Signature and Audit. Certification checks the identification of personal users and agencies using EID (electronic identity) or EBL (electronic business license). Sign uses their signature to get the contracts into effect. Signature is responsible for managing users' signature. Audit is to validate the contracts.
(4)*The Supervisor Module*. This module includes Contract Query, Contract Download, Contract Verification and Contract Record. By Contract Query, users can see the history records of the contracts. Users can download the contracts using Contract Download. Contract Verification is used to compare the uncertain contracts with its fingerprint. Contract Record is used to send the completed contracts to the supervisory authority.

All the four modules contributes to our final system.

5 The Specific program

5.1 Network Topology

Figure 4 shows the network topology. All nodes together form a shared permission blockchain. Institutional participants have choices to become either a full node or a trading node. A full node stores all data, while a trading node chooses to store its own data. All nodes participate in consensus process. Personal users and other institutional investors are connected to shared permission blockchain through gateway provided by Agencies.

The organizational governance of the rest institutions is as follows. Banks act as fund custodian in the business; Law Authority witnesses the transaction and save essential evidence; Market Regulation and Supervisory Authority are in charge of supervision.

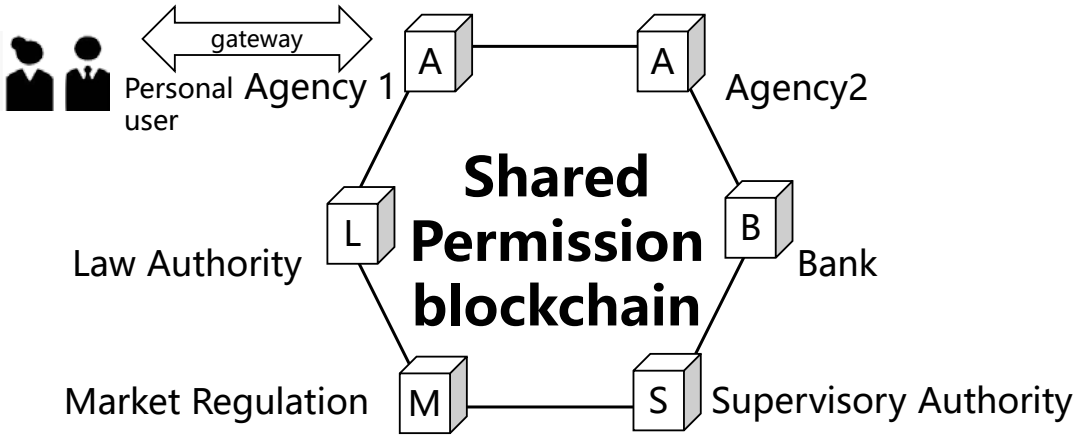

Figure 4: FutureText network topology

5.2 Authentication

Authentication is extremely important in FutureText. It is the key to access the system and service. Most services provided including contract signing, fund transfer and judicial disputes depends on authentication. There are two ways for participants to show self-certificated identity:

For Personal users, electronic identification (EID) is essential when he or she wants to join FutureText. The First Institute of Public Security develops EID. Thus, EID has incomparable advantages over other technologies in terms of authority, security, universality and privacy for it is uniqueness. It perfectly meets our security and undeniable needs in the Privately Offered Fund market.

For institutional users, Electronic Business License (EBL) is essential to identify themselves. The EBL is issued by the Market Regulation. It is also the legal certificate for the market subject to obtain the subject qualification.

The process of authentication is divided into four main steps and is shown in Figure 5:

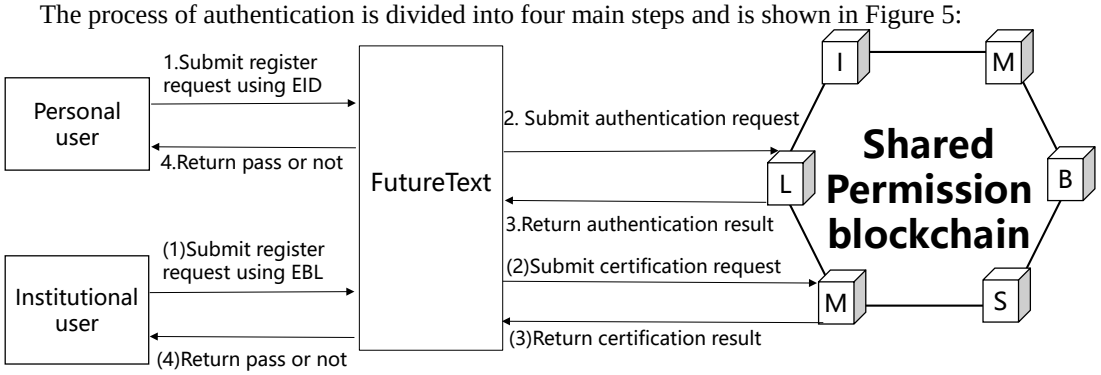

Figure 5: Authentication flow

Step1: Personal user submits EID while institutional user submit EBL.

Step2: FutureText forwards the request to the corresponding authority. EID is sent to the Ministry of Public Security while EBL is sent to the State Administration for Market Regulation.

Step3: The Ministry of Public Security and the State Administration for Market Regulation offer interfaces to validate EIDs and EBLs to see whether they are real or forged. Corresponding results will be sent back to FutureText.

Step4: According to the results, FutureText will generate accounts for those approved. Otherwise, it will close the register request before new information is submitted. Meanwhile, FutureText will inform appliers the results.

5.3 Transaction Data Structure

Blockchain is sequentially linked from the back to the front by blocks containing transaction information. The block containing transaction information is pivotal in FutureText. As shown in Figure 6, the block is divided into Head Info and Transaction List. Some fixed size block metadata is recorded in the Head Info while transactions is recorded in Transaction List.

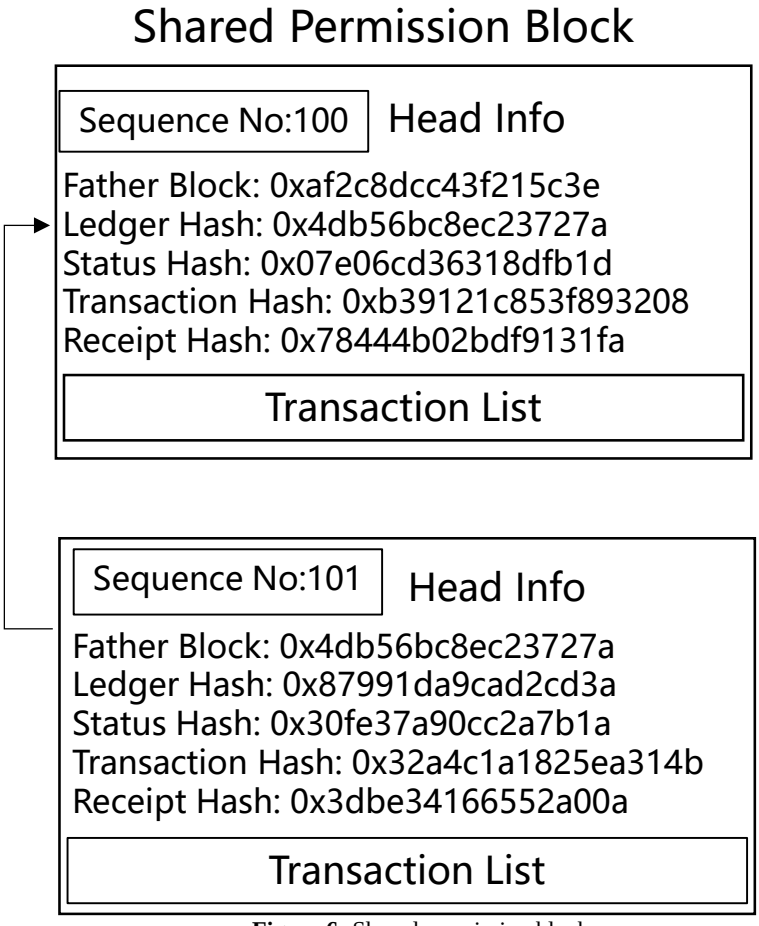

Figure 6: Shared permission block

In Head info, Father Block means the previous block's Ledger Hash. Ledger Hash stands for the hash value of the block. Status Hash represents the status of contract account. Transaction Hash stands for the hash of transaction list included in the block. Receipt Hash is the hash generated by transaction executor.

As for Transaction List, the definition is shown in Table 1.

Table 1: Transaction List definition

Filed name	Description	bytes
Version	Transaction structure definition version information	3
Transaction Hash	Hash identifier produced according to the transaction content	32
Tx Initiator Address	Used to identify the initiator	20
Tx Receiver Address	Used to identify the receiver	20
Smart Contract Call	Call the contract function flag	variable
Timestamp	Approximate time of the transaction	8
Random	Randomly generated integer	8
Signature	Signature information generated by the user	65

5.4 Signature of Tripartite Contracts

In this section, we will discuss how a tripartite contract is completed. Suppose there are three users named A, B and C. A is the manager; B is an investor while C is the custodian. Each user has its own private key and public key. For privacy and safety, the private key is encrypted and saved locally. Users can only decrypted their private key using password. Users' public keys are open across blockchain.

In FutureText, we use two kinds of encryption methods. One is symmetric encryption and the other is asymmetric encryption. Symmetric encryption requires the same key for both encryption and decryption. Asymmetric encryption have two keys in pair. One is public key and the other is private key. We usually use the public key to encrypt and the private key to decrypt.

To denote the workflow of FutureText, here is a case in Figure 7. User A prepares a template contract D. Due to privacy considerations, the contract needs to be encrypted before putting on blockchain. We use symmetric encryption method and the key is generated randomly. The encrypted contract file is ciphertext M. User A hopes User B and C can get the content of the contract. So he encrypts the random number with A's, B's and C's public key to generate three encrypted keys. User A need to sign on the ciphertext M using his private key. After that, user A uploads three encrypted keys, the ciphertext M and File with signature of user A on the blockchain.

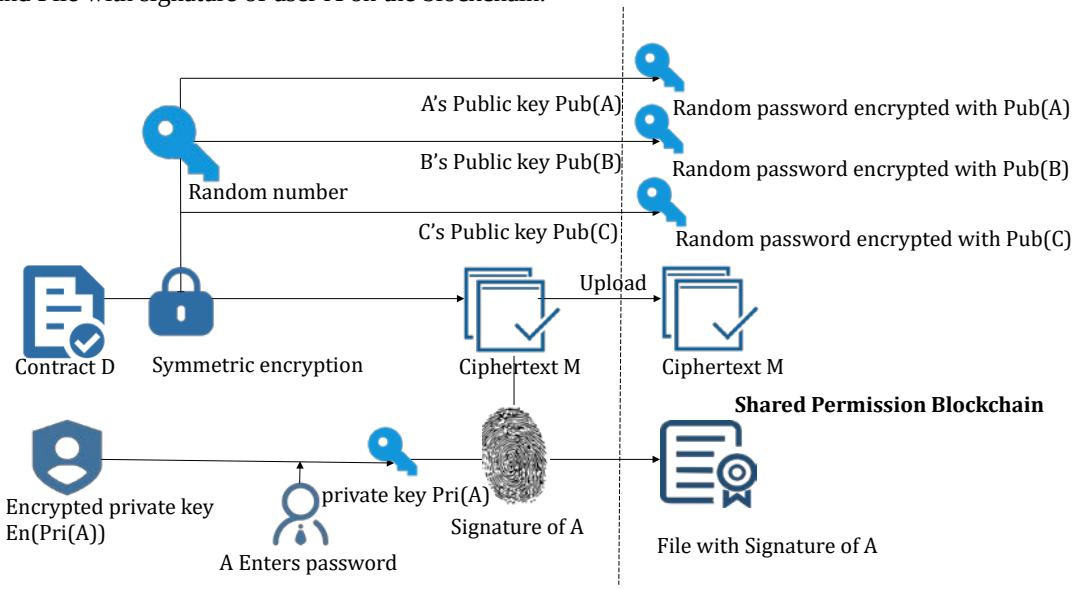

Figure 7: Signature with A's fingerprint

As another contractor of contract D, user B needs to verify the signature of user A after receiving the contract. If the signature is correct, user B will trust that the contract reflects the true will of user A and the contract is not tampered. By using user B's private key, he can get the random number to decrypt the verified ciphertext M. The procedure is shown in Figure 8.

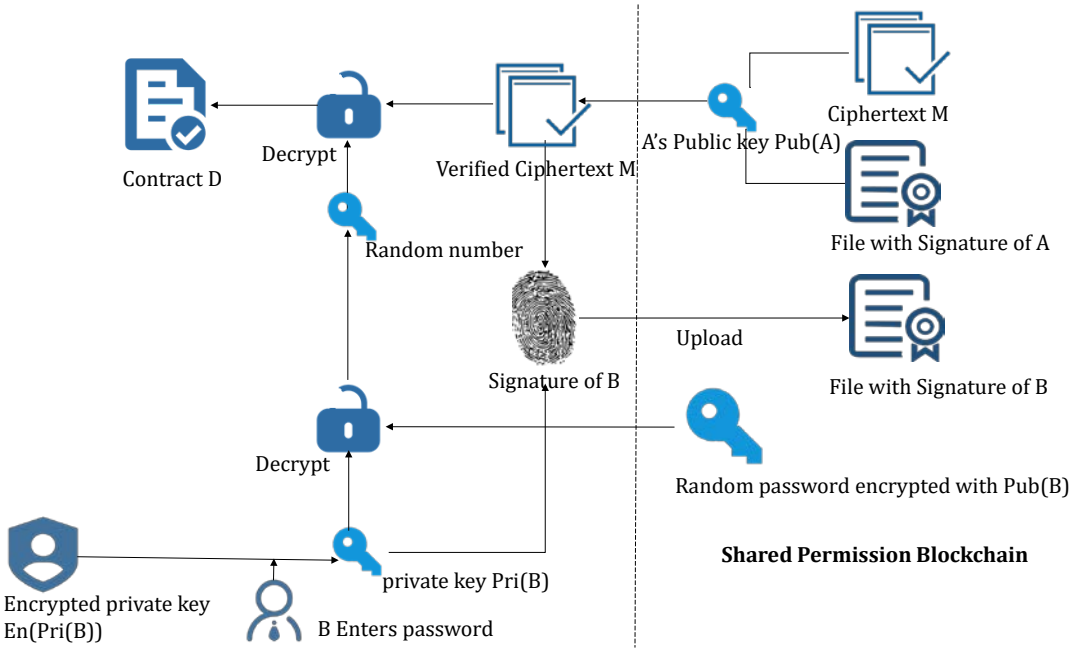

Figure 8: B verifies A's signature

If user B agrees with the content of contract D, then user B will do the similar thing as user A did before, shown in Figure 9. The specific process only needs to be done once again. Finally, we get a file with signature of user A and B, which means both of them have completed the process of signing.

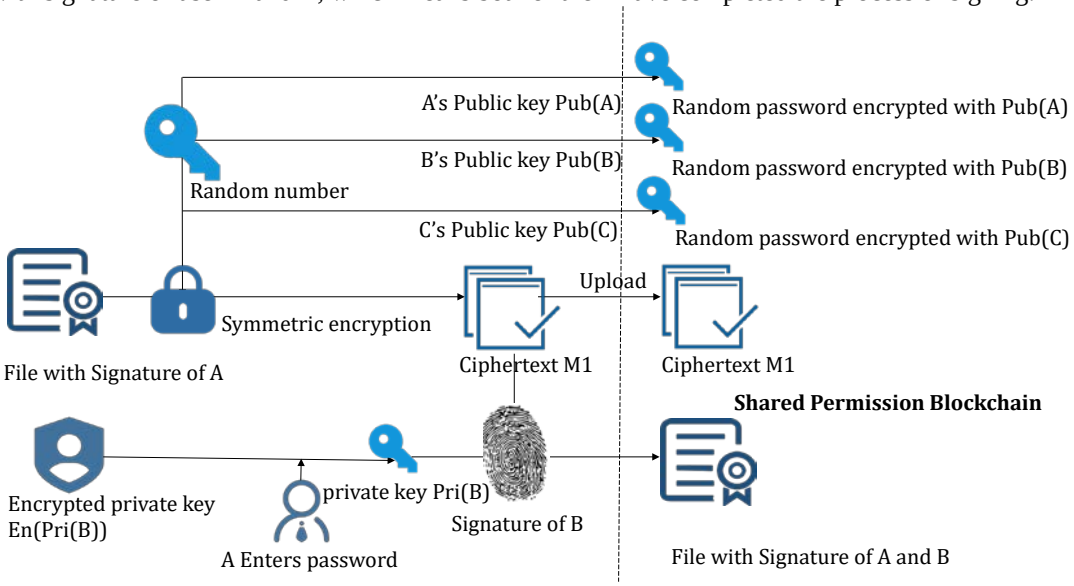

Figure 9: Signature with B's fingerprint

Next, User C signs on the contract. He or she just need to sign the ciphertext M with the private key and add the signature to the file already signed by of A and B. The practice is similar to Figure 8 and Figure 9. For short, we will not discuss with repetition.

At last, we have completed the tripartite contract.

5.5 Contracts Storage

As mentioned in Section 5.4, all contracts are encrypted on blockchain to protect the privacy. Stakeholders may want to search, analysis and review the contract in plaintext. Thus, a heterogeneous storage is designed, as shown in Figure 10. There are six steps to store the contracts.

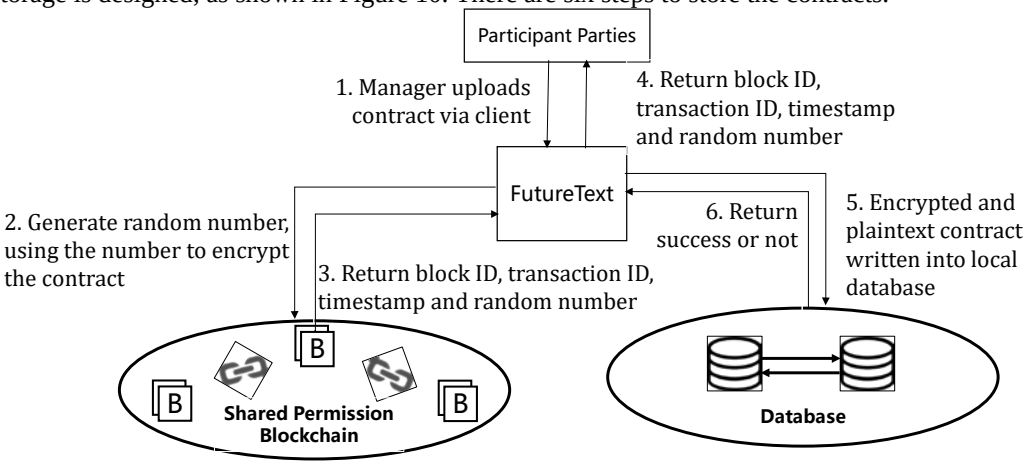

Figure 10: Data storage Flow

Step1: Manager uploads the document via client.

Step2: FutureText generates a random number to encrypt the contract file, and it is synchronized on blockchain.

Step3: FutureText sends back encrypted random number, block ID, transaction ID, and timestamp.

Step4: FutureText forwards the returned information to participants.

Step5: Both encrypted contract and unencrypted contract are stored on local database.

Step6: Database returns the written result.

Through writing on blockchain and the database heterogeneously, FutureText makes sure that data is consistent and stakeholders search the contracts though the local database. All contracts will be submitted to the supervisory authority and kept for another decade even after contacts are expired. Contract details are only available for the signature parties, as well as the supervisory authority.

6 Experiment Evaluation

In this section, we test the efficiency and latency of FutureText. Our experiments use 5 servers, with 4 servers form the shared permission blockchain and 1 server acts as application server. All servers are the same in configuration. All of them have 8 core, 16G memory and 500G hard drive. The shared permission blockchain is Hyperchain 1.6.15. The application server is MySQL 14.14.

The experiments conducted can be divided in two perspectives: transaction per second (TPS) and the latency. The results are as bellows in figure 11. TPS is among 1000 to 2000 pre second and the system latency is around from 400 to 500 milliseconds. Considering the signing frequency of the

Privately Offered Fund contracts, we think it can meet the daily needs. Besides, the delay of FutureText is acceptable.

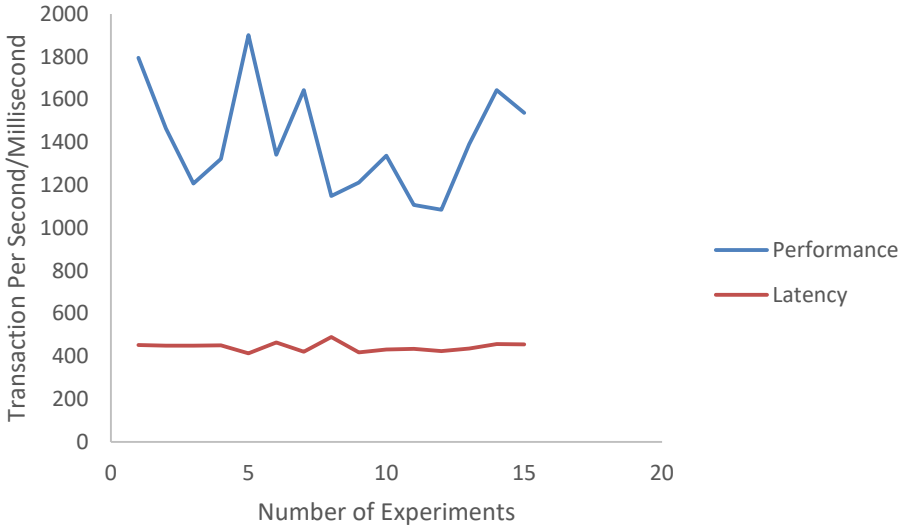

Figure 11: Performance and latency of FutureText

7 Conclusion and Future Work

This paper presents an economical and convenient way for the Privately Offered Fund business based on shared permission blockchain. By introducing an electronic contracts signing and data recording model in FutureText, it alleviates the inefficient problems caused in signing paper printed tripartite contracts in traditional methods. Our system largely reduced the natural and human resource consumption by replacing the paper work with online synchronous cooperation contracts. Meanwhile, we raise an operable authentication method to validate the identification of participants. Finally, a heterogeneous data storage method is applied to provide immediate search and record of signed contracts. The heterogeneous data storage model also relieves the intense query and reply of FutureText.

Never the less, there are still future works for improving. We encountered little problems of storing files on blockchain. It seems becoming a bottleneck for FutureText. Considering the scale and contracts of the Privately Offered Fund market, it implies that large amount of storage usages will be required in the near future. Besides, Hyperchain claims it can reach 10000 TPS in different complexity of the business logic. The performance of FutureText is only one tenth of it.

To solve the potential problems, we may consider partial synchronization and put specific structured data instead of completely raw files onto blockchain. We will also consider optimizing our smart contract to achieve better performance and lower latency.

References

[1]Y. K. Zhou, “On the Formal Issues of Electronic Contract Data Messages,” China Academic Journal Electronic Publishing House, 2019, 06, pp.95.

[2]T. Ge, “Analysis of the burden of proof of online electronic contract signing,” Technology and law, 2018(01), pp.38-44.

[3]X. M. Zhen, “Research on Several Legal Issues of Electronic Contract,” China Academic Journal Electronic Publishing House, 2018, 10, pp.227.

[4]R. Xu, X. J. Meng, F. Ma, et al, “Electronic signing service platform based on tamper-resistant technology,” Computer system application, 2018, 27(4), pp.39-46.

[5]J. L. Cai, W. C. Wang, “Electronic contract signing system,” Information Technology and Standardization, 2018(09), pp.77-80.

[6]Z. C. Yin, H. Li, “Electronic contract solution based on Hyperledger technology,” Modern computer, 2018(04), pp.86-89.

[7]J. T. Gu, X. D. Zhu, “Designing and Implementation of an Online System for Electonic Contract Negotiation Based on Electronic Signature,” Journal of Software, 2014, 9(12), pp.3020-3027.

[8]A. Burunova, A Ponomarev, N. Teslya, “Enactable Electronic Contracts In E-Commerce: Models, Technologies And Architectures,” Proceeding of the 24th Conference of FRUCT Association.