SDSBT: A Secure Multi-party Data Sharing Platform Based on Blockchain and TEE

Hong Lei[1], Yun Yan[1], Zijian Bao[1], Qinghao Wang[1,2(✉)], Yongxin Zhang[1,2], and Wenbo Shi[2]

[1] Oxford-Hainan Blockchain Research Institute, Chengmai 571924, China
{leihong,yanyun,zijian,qinghao,yongxin}@oxhainan.org
[2] Department of Computer Science and Engineering, Northeastern University, Shenyang 110001, China
shiwb@neuq.edu.cn
https://www.oxhainan.org

Abstract. With the rise of big data analytics and artificial intelligence, an increasing number of enterprises and individuals are concerned about the security and privacy of the shared data. However, it is still challenging to achieve a data sharing scheme, which meets the security, privacy, security, and credibility requirements. This paper proposes SDSBT, a multi-party data sharing platform based on blockchain and the trusted execution environment (TEE), which effectively and securely realizes the data sharing among multiple parties. SDSBT achieves the properties including privacy-preserving, identity authentication, application security, and accountability. The security analysis and experimental results show that the scheme is secure and practical.

Keywords: Blockchain · TEE · SGX · Data sharing · Privacy protection

1 Introduction

With the development of big data analytics and artificial intelligence, data is becoming more and more valuable and has become the core asset of some enterprises and government agencies. One of the fundamental enabling components for the big data analytics and artificial intelligence is the abundance of data. Thus, it is essential to share data between multiple parties for the abundance of data [1,2]. However, as more information on individuals is shared and analyzed, data providers are getting increasingly concerned about data security and privacy issues. How to effectively coordinate all parties to complete data sharing with a secure and privacy-preserving way is one of the current research hotspots [3–11].

This study is supported by Oxford-Hainan Blockchain Research Institute, the National Science Foundation of China (No. 61472074, U1708262) and the Fundamental Research Funds for the Central Universities (No. N172304023).

J. Cheng et al. (Eds.): CSS 2020, LNCS 12653, pp. 184–196, 2021.
https://doi.org/10.1007/978-3-030-73671-2_17

Recently, many scholars propose schemes to solve the security problems in data sharing. Zhao et al. [7] proposed a data sharing model for the Internet of things(IoT) scenario, which used the attribute-based encryption to achieve fine-grained access control of ciphertext and guarantee the data privacy. However, the efficiency and security of the scheme need to be further improved. In [8], the authors implemented a lightweight data sharing platform, which used the data integrity verification to enhance security. But the scheme did not consider the security of the data processing. Wu et al. [9] proposed a human intelligence, artificial intelligence, and organizational intelligence (HAO) governance method to achieve the requirements of data standard governance. However, the security properties of data storage were not considered.

Blockchain is a public and decentralized ledger, which is tamper-proof and traceable. One rising way is to use the blockchain technology to enhance the security of data sharing. Wang et al. [10] proposed a data sharing platform, which employs the tamper-proofing and traceability of the blockchain to ensure the correctness of data. In [11], Wang et al. construct a blockchain-based data sharing network, which implements the secure data sharing. However, both of the schemes store the source data in the blockchain, which is visible to everyone. Thus the data privacy can not be protected.

The trusted execution environment (TEE) technology implements the secure container to protect the integrity and confidentiality of the internal data [12], which can provide powerful support for the blockchain-based data sharing schemes. In this paper, we designed a multi-party data sharing platform, which meets the privacy, security, and credibility requirements of the data sharing, by combining TEE and blockchain. We build the data sharing platform based on TEE and run the data computation in the secure container to protect the data privacy. The secure container can also guarantee the correctness of the internal computation, thus the security of data sharing is ensured from the hardware level. Moreover, we store the data sharing log in the blockchain to provide a basis for data confirmation and to realize the functions of retrospective and accountability afterward.

In summary, in this paper, we make the following contributions:

1. We propose a data sharing platform SDSBT based on blockchain and TEE, which achieves privacy-preserving, identity authentication, application security, and accountability properties.
2. Through simulation and analysis of experiments, the feasibility of the scheme in this paper is confirmed.

The structure of the paper is as follows: in Sect. 2, the related background is provided, including TEE, Intel SGX Software Guard Extensions (SGX), and Blockchain technology. In Sect. 3, we describe the current system architecture, threat model, and design goals. In Sect. 4, we propose the system design. In Sect. 5, the security of the platform is analyzed. In Sect. 6, we conduct the corresponding experimental analysis. The paper is concluded in Sect. 7.

2 Background

To better understand our scheme, the related backgrounds are presented, including the basic concept of TEE, SGX, and blockchain technology.

2.1 Trusted Execution Environment (TEE)

The trusted execution environment (TEE) technology can provide the secure container to prevent potentially malicious users from controlling or observing the internal data [12]. To achieve the secure container, TEE needs to guarantee the data isolation between the trusted environment and the normal environment. A common scheme is providing the physical memory isolation and the extra access checks, so that the normal environment cannot directly access the data in the trusted environment. Currently, the popular TEE solutions include SGX(SoftwareGuard Extensions) based on Intel x86 architecture, TrustZone based on ARM Architecture, and MultiZone [13] based on open-source framework RISC-V.

At present, TEE [14] technology is used in many fields to improve privacy protection and security properties. Chen et al. [15] used TEE to defense causative attacks and achieve data privacy in federated learning systems. Sebastian et al. [16] use TEE technology to enhance the security of IoT devices and provide the implementation of the proposed architecture on the platform supporting ARM TrustZone.

2.2 Intel Software Guard Extensions

Intel software guard extensions (SGX) is one of the most popular TEE products in commodity CPUs, which is employed in various fields [17–19]. It provides the new CPU instructions on the Intel CPU architecture to guarantee the integrity and confidentiality for the security-sensitive computing performed on the SGX-enabled machine where the privileged softwares (e.g., kernel, hypervisor, and so on) are malicious [20].

Essentially, the new instructions are coupled with a pre-partitioned memory called enclave page cache (EPC), which is the secure memory protected by the hardware. Users can employ the instructions to create the secure container called enclave for their applications in EPC [21], and the access to the enclave will be performed additional checks to protect the security. To ensure the reliability of the SGX-enabled platform, SGX implements the remote attestation to prove to the remote parties that the particular application is loaded in the enclave, and the enclave is running in the real SGX-enabled platform. Comparing to the other TEE schemes, the trusted computing base (TCB) of SGX is smaller, which only contains the CPU and EPC. Any privileged software, such as OS, hypervisor, BIOS, SMM, etc., is not included in its TCB.

Note that we employ SGX as the TEE instance to implement the simulation experiment of our schemes. Thus the characteristic of TEE is obeyed to SGX in the paper.

2.3 Blockchain Technology

Blockchain is a public, distributed ledger, which can record transactions in a verifiable and immutable way. It is jointly maintained by the nodes in the blockchain system, and the nodes constitute a vast P2P network. The consistency of the ledger is guaranteed by the consensus algorithm, which is the basis for quickly reaching a consensus on the blockchain data among highly dispersed nodes. The consensus algorithm allows the transactions to be securely stored and verified without any centralized authority. A blockchain is a growing list of records, called blocks, that are linked using cryptography technology. Each block contains a hash value of the previous block, a timestamp, and transaction data (generally represented as a Merkle tree).

3 Problem Statement

3.1 System Model

As shown in Fig. 1, the blockchain-based TEE secure multi-party data sharing platform is mainly divided into four parts: data providers, data users, cloud servers, and a blockchain.

- **Data providers.** The data provider uploads data to cloud servers and shares it into the data sharing platform.
- **Data users.** The data user uses the data sharing platform to access the desired data.
- **Cloud servers.** The cloud server is responsible for the secure transmission of data, secure calculations, and interaction with the blockchain.
- **Blockchain.** The blockchain and the associated smart contracts are used for storing transaction information and driving the transaction process.

3.2 Threat Model

In our assumption, the data user is an unreliable role. It will try to steal the real data. Besides, the platform is also not enough security. It may be compromised by the adversary. Moreover, we assume the TEE is normally trustworthy, i.e., the adversary cannot compromise it. The handled data can be protected by the enclave. We also assume that the adversary cannot control the blockchain, such as tampering with the data, blocking the consensus algorithm.

3.3 Design Goals

As SDSBT is designed to provide a trusted sharing environment for users, it should meet the following goals: privacy-preserving, identity authentication, application security, and accountability.

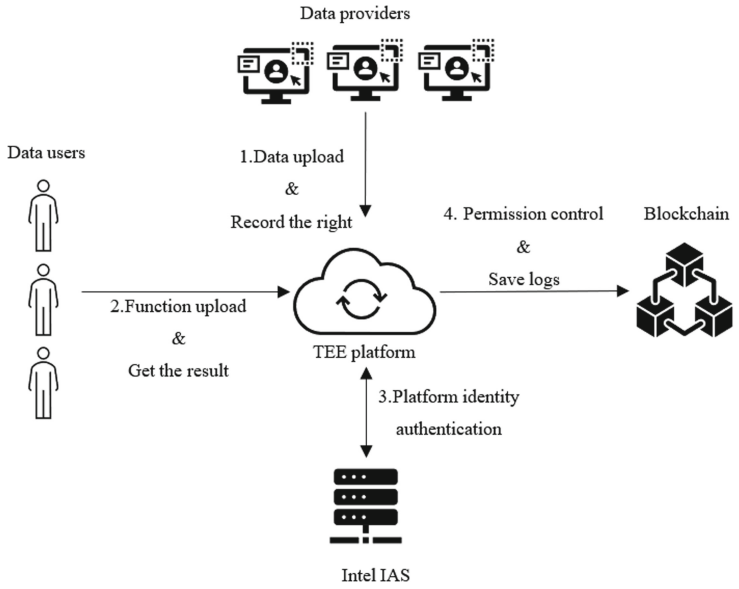

Fig. 1. System model

(1) *Privacy-Preserving*: during the sharing process, the platform and data users cannot infer the user's real data.
(2) *Identity Authentication*: in our scheme, unauthorized users cannot participate in the sharing process.
(3) *Application Security*: it is difficult for the adversary to steal or corrupt the sharing data in runtime.
(4) *Accountability*: in our scheme, once a role violates the protocol, it will be found and blamed.

4 SDSBT System Design

In this section, we will systematically elaborate on the infrastructure and interaction process of our platform. For the convenience of illustration, as well as given the fact that SGX is currently the most widely used TEE technology, we will adopt SGX to replace the term TEE in this section. We use SGX to construct the main part of the platform and design five layers. At the same time, SDSBT strictly controls user access rights. It provides users with access controls at the contract layer and read permissions at the data sharing layer, which reduces the risk of the platform.

4.1 The Infrastructure of SDSBT

As shown in Fig. 2, the overall architecture can be divided into five layers: contract layer, data transmission layer, data operation layer, data sharing layer, and storage layer.

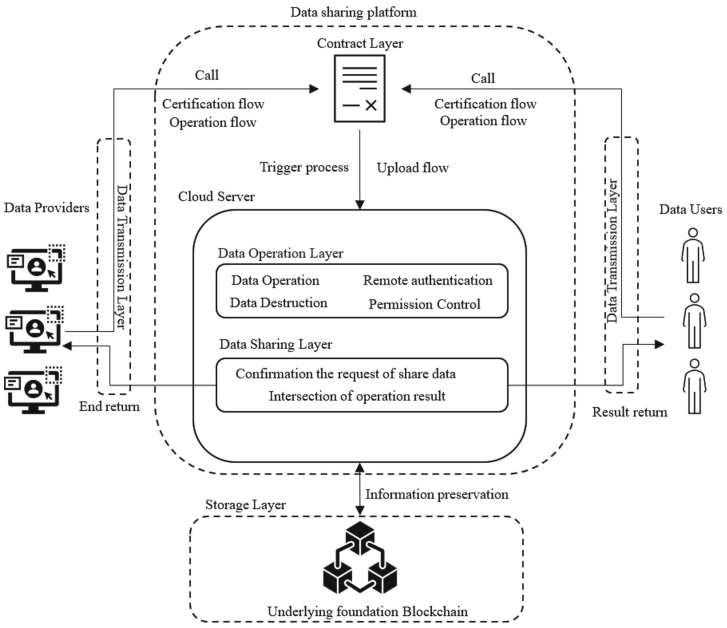

Fig. 2. The platform infrastructure.

Contract Layer. The TEE contract layer mainly implements the driving function of the data sharing process. The data sharing parties complete the data sharing process by calling the contract. The contract layer mainly provides data/code upload functions, platform authentication, and authentication result generation and triggers the platform for data processing.

Data Transmission Layer. In the entire data sharing platform, the basic step is to transmit the local data of the data owner to the cloud platform in a secure and reliable manner, and to ensure that there will be no private copy or storage of data on the platform level during the transmission process, or data leakage caused by malicious third-party attacks.

To achieve this, we first use the enclave of the cloud platform to construct a TLS secure transmission channel from the cloud to the data owner. Then, we conduct the hash value of the code to build the secure channel, and store it in the blockchain to guarantee the data owner can check the authenticity of the data security transmission function.

Data Operation Layer. The data operation layer is the core of the entire secure multi-party data sharing platform. It is a cloud data computing sub-platform built by the SGX server cluster to fulfill the functional requirements of data users.

The data consumer needs to upload the verified function code into the data operation layer. Next, the data operation layer puts the code and the original data into the SGX for execution to obtain the operation result, and then deletes the original data after the execution.

Data Sharing Layer. The data sharing layer mainly completes the sharing process between users and users, ensures users complete the whole sharing strategy according to the established sharing process, and at the same time sends instructions for storing data in the blockchain to the storage layer.

Storage Layer. The storage layer is the underlying blockchain, which utilizes its secure and traceable feature to save the transaction information on the chain and provide the transaction verification mechanism.

4.2 Interactive Process of SDSBT

In this sub-section, we propose the interaction flow in Fig. 3. There exist data providers and data users, denoted as s_i and u_i $(i > 0)$, respectively. We further indicate the remote authentication result by r_i, the original data set by D_i, the function code for the data user by P_i, the basic information of the platform hardware by O_i, the result of the operation by Ans, the end of the operation by f_o, the end of the transaction for the data user by f_{tu}, the end of the transaction for the data supplier by f_{ts}, the overall transaction information by inf, and the contract execution operation by Con.

User Registration Process (Step 1). The data user registers on the data sharing platform, including data attributes and user information. The platform assigns a user ID and a monotonic counter to the registered data owner and maintains a table in the database as a retrospective basis.

Function Upload and Attestation Process (Step 2 – Step 6). The data user sends the function code to the contract layer and calls the command to trigger the attestation in the contract. At this time, the contract layer drives the SGX data operation layer for local and remote attestation to the authentication agency. Then the remote authentication agency calls IAS(Intel Attestation Server) to complete the remote attestation of the device and function code, and returns the attestation result to the contract driver layer. The data sharing parties confirm the attestation result. Then the verification process can be expressed as follows, where the n means the total number of O_i and the m means the total number of P_i:

$$\sum_{i=1}^{n}\sum_{j=1}^{m}(O_i + P_j) \xrightarrow[ISA]{Con} \sum_{k=1}^{t} r_k, i \in (0,n], j \in (0,m], t \in (0,t] \tag{1}$$

$$\sum_{k=1}^{t} r_k \xrightarrow{Con} s_i \tag{2}$$

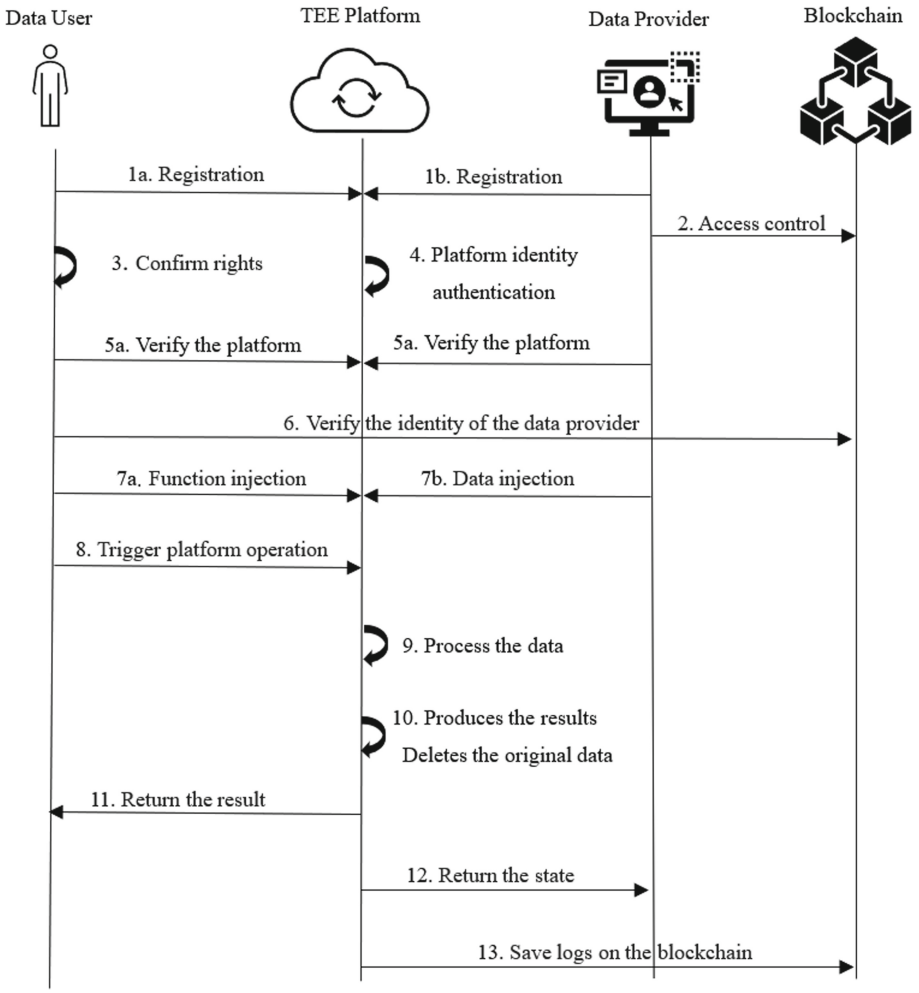

Fig. 3. Diagram of platform interaction process.

Data Upload and Computation Process (Step 7 – Step 9). After receiving the authentication result, the data sharing parties upload the corresponding data and code by calling the contract. The contract layer passes the code and data to the SGX data operation layer. After that, the data operation layer performs data processing and ensures that original data is deleted in time during the processing. Then the data upload and computation process can be expressed as:

$$\sum_{i=1}^{n} D_i \xrightarrow{Con} O, i \in (0, n] \tag{3}$$

$$\sum_{i=1}^{n}\sum_{j=1}^{m}(D_i + P_j) \xrightarrow{O} Ans + f_o, i \in (0,n], j \in (0,m] \tag{4}$$

$$f_o \xrightarrow{Con} u_i \& s_i \tag{5}$$

Data Share and Destruction Process (Step 10 – Step 12). After the data operation is completed, the data sharing layer will return the data operation result to the data consumer and the data completion status to the data provider. After that, the data operation layer and the data sharing layer will automatically delete the intermediate data.

Then the transaction process can be expressed as:

$$f_{ts} \xrightarrow{s_i} Con \tag{6}$$

$$f_{tu} \xrightarrow{u_i} Con \tag{7}$$

$$f_{tu} + f_{ts} \xrightarrow[Ans]{Con} u_i \tag{8}$$

Certificate Preservation Process (Step 13). After the data sharing layer completes the transaction recording, it will save the agreed content of the operation information on the chain so that the platform users can trace back the related information when needed.

Then the process of saving the certificate can be expressed as follows, where the "$Hash$" means the hash function:

$$f_{tu} + f_{ts} \xrightarrow[Hash]{T} inf \tag{9}$$

$$inf \xrightarrow{Hash} P \tag{10}$$

5 Security Analysis

During the process, the data owner needs to ensure that the original data cannot be obtained by the data user without authorization [21]. As a data sharing platform, our system can provide measures for owners to secure data, including data privacy-preserving for data owners, identity authentication for data users, application security for sharing platform, and accountability for malicious users.

Privacy-Preserving. In our system, data users cannot access the real data without procession, such as encryption, aggregation. We isolate the sharing platform and the data computing platform. The former is responsible for interacting with the data consumer and the latter is used to operate on target data as required by the data user, e.g., to return a result of aggregation. As a result, the data user cannot get the real data.

Identity Authentication. Our system provides an identity authentication mechanism for both parties of the process. In the first step, all the users should register in the sharing platform, where each user owns a unique identity in the system. In the next sharing process, each communication data is attached to the ID of a user. Data owners and data users can confirm the identity of the other party based on the ID.

Application Security. To achieve the security of the application, we adopt the TEE to operate the data. The pre-made code and uploaded code on the platform are displayed and attested to provide code security for the data owner. Then, all the computing resources and computing objects are placed in the TEE environment to ensure the safe and reliable operation of data and code, thereby ensuring the security of the application.

Accountability. To further implement the punishment for malicious users, we take advantage of the blockchain, which is not tampered with. The entities' operations during the sharing process are recorded by the blockchain system. Once someone breaks the sharing rules, e.g., delay the submission time, others can use the records in the blockchain to claim the malicious.

6 Experimental Evaluation

We implement the simulation experiment to evaluate the performance of the cloud server, which is the only part with "nonlinear" performance overhead in our scheme (we will explain that why the performance overhead is "nonlinear" in the following paragraph). Specifically, we run a typical calculation that searching for target data in the datasets of different sizes. The cloud server runs on the SGX-enabled PC with Ubuntu 16.04 LTS operating system, a 3.6 GHz Intel(R) Core(TM) CPU i3-9100F, and 8 GB RAM. The source dataset of the experiment includes the metadata of 1.7 million arXiv scholarly papers gathered by Cornell University [22].

The secure memory size of TEE is limited, and the oversubscription of memory may result in extra overhead. For example, SGX only supports secure memory that is smaller than 128 MB, and it requires additional operations to oversubscribe the secure memory by evicting and loading enclave pages securely, which will result in significant overhead. Thus, the calculation of different data volumes in the TEE may bring "nonlinear" performance overhead. To test the performance overhead caused by the use of TEE, we compare the computation time of the searching operations executed in the enclave with the time of the same operations performed out of the enclave in different dataset sizes (e.g., 1M, 10M, 100M, 500M, and 1G) derived from the arXiv metadata. Figure 4 shows that the former is slightly more than the latter in 1M, 10M, 100M sizes, which is because the programs in the enclave need to execute the additional encryption and decryption operations. Moreover, with the 500M and 1G sizes, the former is obviously more than the latter. It is because processing the excessive data will oversubscribe the secure memory, which will generate frequent scheduling

operations to bring the extra performance overhead. Thus, a possible future work is that employing the collaboration of multiple cloud servers to reduce the computational overhead of the single node.

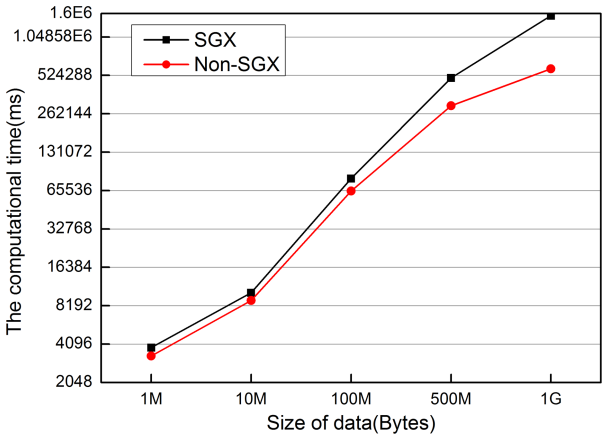

Fig. 4. Experimental results

7 Conclusion

In order to address the security challenges in existing blockchain-based data sharing schemes, we propose a secure multi-party data sharing platform combining blockchain and TEE (i.e., SDSBT). In SDSBT, the tamper-proof features of smart contracts not only ensure the security of the platform itself but also provide users with a reasonable code review path. The operational security of the data sharing platform is guaranteed through the trusted TEE server cluster and the overall performance of the platform is improved through separating the transaction layer and the operation layer. TEE improves the security of our system, but it also imposes certain performance limitations. Through experiments, we have shown that the performance difference between using TEE and not using TEE is significant for a large amount of data due to the limited secure memory of SGX.

Future Work. With the increasing market demand for data sharing, the demand for secure data sharing platforms and the platform visits will continue to increase. Under such circumstances, we need to consider more about the security and usability of the platform.

In terms of the platform security, we mainly consider further enhancing the security capabilities of TEE, due to it suffers the side-channel attacks. For example, we can consider using Oblivious Random Access Machine (ORAM) to further enhance memory protection [23] to resist side-channel attacks. In addition,

it is also promising to consider the fusion of trusted computing, secure multi-party computing and SGX [24–26] to achieve stronger security.

In terms of the platform usability, from the perspective of the arnchitectural design of the platform, in order to meet the requirements of security, computing and usability of the users in the future, we will optimize the interactive architecture of the platform and further distinguish on-chain management and off-chain computation functions to enhance its security and efficiency.

References

1. Lu, Y., Huang, X., Zhang, K., Sabita, M., Zhang, Y.: Blockchain empowered asynchronous federated learning for secure data sharing in internet of vehicles. IEEE Trans. Veh. Technol. **69**(4), 4298–4311 (2020). https://doi.org/10.1109/TVT.2020.2973651
2. Imran, M., Ian, Z., Mehran, A., Justin, L., Ni, W.: PrivySharing: a blockchain-based framework for privacy-preserving and secure data sharing in smart cities. Comput. Secur. **88**, 101653 (2020). https://doi.org/10.1016/j.cose.2019.101653
3. Hoon, W., Geong, S., Xu, J., Varsha, C.: PrivateLink: privacy-preserving integration and sharing of datasets. IEEE Trans. Inf. Forensics Secur. **15**, 564–577 (2020). https://doi.org/10.1109/TIFS.2019.2924201
4. Ma, H., Zhang, R., Yang, G., Song, Z., He, K., Xiao, Y.: Efficient fine-grained data sharing mechanism for electronic medical record systems with mobile devices. IEEE Trans. Dependable Secur. Comput. **17**(5), 1026–1038 (2020). https://doi.org/10.1109/TDSC.2018.2844814
5. Shen, J., Zhou, T., He, D., Zhang, Y., Sun, X., Xiang, Y.: Block design-based key agreement for group data sharing in cloud computing. IEEE Trans. Dependable Secur. Comput. **16**(6), 996–1010 (2019). https://doi.org/10.1109/TDSC.2017.2725953
6. Li, H., Zhu, H., Du, S., Liang, X., Shen, X.: Privacy leakage of location sharing in mobile social networks: attacks and defense. IEEE Trans. Dependable Secur. Comput. **15**(4), 646–660 (2018). https://doi.org/10.1109/TDSC.2016.2604383
7. Zhao, Z., Wang, J., Zhu, Z., Sun, L.: Attribute-based encryption for data security sharing of internet of thing. J. Comput. Res. Dev. **56**(6), 1290–1301 (2019). https://doi.org/10.7544/issn1000-1239.2019.20180288
8. Lu, X., Cheng, X.: A secure and lightweight data sharing scheme for internet of medical things. IEEE Access **8**, 5022–5030 (2020). https://doi.org/10.1109/ACCESS.2019.2962729
9. Wu, X., Dong, B., Du, X., Yang, W.: Data governance technology. Ruan Jian Xue Bao **30**(9), 2030–2856 (2019). https://doi.org/10.13328/j.cnki.jos.005854
10. Wang, J., Wei, S., Dai, K.: Research on open data sharing system based on blockchain in the area of financial services system for science and technology. Modern Comput. 22,52–58+78 (2018). https://doi.org/10.3969/j.issn.1007-1423.2018.22.01
11. Wang, J., Gao, L., Dong, A., Guo, S., Chen, H., Wei, X.: Block chain based data security sharing network architecture research. J. Comput. Res. Dev. Ruan Jian Xue Bao **54**(4), 742–749 (2017). https://doi.org/10.7544/issn1000-1239.2017.20160991
12. TEE Committee (formerly Device Committee). https://globalplatform.org/technical-committees/trusted-execution-environment-tee-committee/

13. Newsome, T.: Megan Wachs. RISC-V External Debug Support, SiFive (2019). https://riscv.org/specifications/debug-specification/
14. Yan, Z., Venu, G., Zheng, Q., Wang, Y.: Access special section editorial: trusted computing. IEEE Access. **8**, 25722–25726 (2020). https://doi.org/10.1109/ACCESS.2020.2969768
15. Chen, Y., Luo, F., Li, T., Xiang, T., Liu, Z., Li, J.: A training-integrity privacy-preserving federated learning scheme with trusted execution environment. Inf. Sci. **522**, 69–79 (2020). https://doi.org/10.1016/j.ins.2020.02.037
16. Jonathan, D., Utkarsh, A., Ali, T., Adam, H.: DER-TEE: secure distributed energy resource operations through trusted execution environments. IEEE IoT J. **6**(4), 6476–6486 (2019). https://doi.org/10.1109/JIOT.2019.2909768
17. Shen, T., Jiang, J., Jiang, Y., et al.: DAENet: making strong anonymity scale in a fully decentralized network. IEEE Trans. Dependable Secure Comput. (2021)
18. Wu, L., Cai, H.J., Li, H.: SGX-UAM: a secure unified access management scheme with one time passwords via Intel SGX[J]. IEEE Access **9**, 38029–38042 (2021)
19. Schwarz, F., Rossow, C.: SENG, the SGX-enforcing network gateway: authorizing communication from shielded clients. In: 29th USENIX Security Symposium (USENIX Security 20),pp. 753–770 (2020)
20. Why use Arm architecture? Performant. Efficient. Compatible. https://developer.arm.com/architectures
21. Intel Corporation. Intel® software guard extensions (Intel® SGX). Intel Labs (2013). https://software.intel.com/sgx
22. ArXiv dataset and metadata of 1.7M+ scholarly papers across STEM. https://www.kaggle.com/Cornell-University/arxiv
23. Adil, A., Kyungtae, K, Muhammad, S., Byoungyoung, L.: Obliviate: A Data oblivious filesystem for intel SGX. In: 25th Annual Network and Distributed System Security Symposium, NDSS, San Diego, California, USA, 18–21 February 2018
24. Sasy, S., Gorbunov, S.. Fletcher, C.W.: ZeroTrace: oblivious memory primitives from Intel SGX. In: 25th Annual Network and Distributed System Security Sy.mposium, NDSS, San Diego, California, USA, 18–21 February 2018
25. Bahmani, R., et al.: Secure multiparty computation from SGX. In: Kiayias, A. (ed.) FC 2017. LNCS, vol. 10322, pp. 477–497. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-70972-7_27
26. Zegzhda, D.P., Usov, E.S., Nikol'skii, A.V., Pavlenko, E.Y.: Use of Intel SGX to ensure the confidentiality of data of cloud users. Autom. Control Comput. Sci. **51**(8), 848–854 (2017). https://doi.org/10.3103/S0146411617080284