Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Data resource protection based on smart contract

*Wei Xiong, Li Xiong**

Department of Information Management, School of Management, Shanghai University, Shanghai, China

ARTICLE INFO

Article history:
Received 3 April 2020
Revised 30 July 2020
Accepted 16 August 2020
Available online 22 August 2020

Keywords:
Data resource protection
Ethereum
Blockchain
Smart contract
Machine learning

ABSTRACT

In this paper, a data resource protection solution is proposed to eliminate the potential risk of data owners reselling others' data resources. By utilizing the key functions of Ethereum blockchain and smart contract, a data-for-sale mechanism is built. By this mechanism, data-for-sale information is received and processed, the information is broadcast to the nodes in the blockchain, and the feedback of the nodes to the information is handled. By introducing large-margin multi-task metric learning, a dispute resolution mechanism is constructed to solve data-for-sale disputes. By designing "data-for-sale", "access to market", "dispute resolution" algorithms, data resource protection is realized. By proposing sequence diagrams and algorithms, smart contract is finished. Finally, according to the solution, the test and validation of the smart contract is completed. The code for the smart contract and ABI interface is available on GitHub.

1. Introduction

With the development of Internet of Things and other information technologies, all kinds of data are collected by the technologies (Sun et al., 2016). The sharing and trading of the data is the trend of the times and the inevitable demand of the market. An increasing number of data trading centers (e.g., CitizenMe, DataExchange, Factual and Terbine) are already offering B2C or C2C data trading services. They are the counterparts of physical commodities trading centers (e.g., Amazon and Ebay) (Jung et al., 2018). Although data trading is conducive to the sharing and utilizing of the data, and can generate considerable profits, there are potential risks in the process of data trading, such as data replication, data retention and data resale. These risks lead to the unwillingness of entities with data to participate in data trading (Yang, 2016). One of the main risks is that data purchasers can update the purchased data by adding data with different labels. Then they can resell the purchased data. Because data replication is undifferentiated and data delivery is almost cost-free, the profit from reselling data is considerable relative to the profit from reselling physical goods. If data trading centers or data purchasers resell the acquired data, they can get bigger sales volume and profits. Concerns arise at the data trading center side: data owners fear that data trading centers may disclose or resell the hosted data. On the other hand, concerns arise at the data purchaser side: data owners worry that data purchasers may resell the purchased data. The first problem can be solved because data trading is decentralized by blockchain (Nakamoto, 2020) and smart contract (Szabo, 2020), and the data is stored in IPFS (IPFS is the Distributed Web, 2020), instead of being hosted in the data trading center (Xiong and Xiong, 2019). Thus, the data trading center can not access the data, nor can it disclose or resell the data. The second problem is difficult to solve, because data purchasers cannot accept being monitored. Firstly, it is illegal to monitor the behavior of data purchasers. This violates the privacy of data purchasers. Secondly, when using the Internet, data purchasers will not

* Corresponding author.
E-mail address: xiongli8@163.com (L. Xiong).
https://doi.org/10.1016/j.cose.2020.102004

voluntarily install monitoring devices to ensure their own security and user experience.

To eliminate the risk of data resale, a data resource protection solution based on smart contract is proposed. The solution mainly includes two mechanisms, namely, data-for-sale mechanism and dispute resolution mechanism. Note that the centralized data-for-sale mechanism is the reason why data trading centers may leak or resell others' data. The lack of dispute resolution mechanism is the reason why data purchasers may resell others' data. The non-transparency of the centralized data-for-sale mechanism and the lack of a dispute resolution mechanism have caused data owners to have distrust and unreliable concerns about the sharing and trading of data. Hence, a decentralized data-for-sale mechanism is built by blockchain and smart contract. On the one hand, the blockchain is a decentralized distributed ledger. It has some characteristics, such as immutability, tamper-proof, traceability, event system and tamper-proof ordered log (Hasan and Salah, 2018). It is worth noting that blockchain-based solutions are widely used, including finance (Treleaven et al., 2017), health care (Azaria et al., 2016), intrusion detection (Meng et al., 2018), food industry (Tian, 2017), supply chain management (AlTawy et al., 2017) and other fields. On the other hand, as a computer program, the smart contract can effectively receive, process, store and send information. Furthermore, as a kind of code, the smart contract can control assets and respond to information received from the outside (Cruz et al., 2018). The smart contract has four main characteristics: (1) Consistency. The smart contract needs to comply with existing legal provisions; (2) Observability. The observability can ensure that the smart contract is effectively regulated; (3) Compulsory. With this feature, even if the compulsory force of the law does not interfere, some breaches of contract can be avoided; and (4) Access control. This is mainly used for third-party verification (He et al., 2018). In addition, large-margin multi-task metric learning is used to construct a dispute resolution mechanism. The large-margin multi-task metric learning is a machine learning algorithm. It is composed of large-margin nearest neighbor classification and multi-task learning paradigm. It can be used to distinguish data with similar labels from data with different labels. The imposters in the data will be isolated to determine the similarity of the data (Parameswaran and Weinberger, 2010).

There are some challenges in designing the data resource protection solution. (1) How to use blockchain and smart contract to realize data resource protection after third-party trading platforms are decentralized. The difficulty lies in how to prevent data resale. (2) When data-for-sale disputes occur among data owners, data trading centers can not act as arbitrators to deal with the disputes, because they are the trading participants and may collude with other trading participants. The difficulty is that a trusted entity needs to be introduced to deal with data-for-sale disputes after data trading centers are decentralized. To solve these challenges, the data-for-sale and dispute resolution mechanisms are established. The "data-for-sale", "access to market" and "dispute resolution" algorithms are designed. The arbitration institution is introduced by assuming that the arbitration institution is a trusted entity dealing with data-for-sale disputes. By introducing large-margin multi-task metric learning, the data resource protection solution can effectively prevent data resale. If there is no objection to the data-for-sale of the data owner, the data-for-sale will be sold through the blockchain network.

The main contributions of this paper can be summarized as follows:

1. A new data-resource-protection solution is proposed. Data resources are protected by building data-for-sale and dispute resolution mechanisms.
2. Data-for-sale request, data-for-sale dispute resolution, and automatic refund and compensation are realized by designing "data-for-sale", "access to market" and "dispute resolution" algorithms.
3. The implementation and testing details of the smart contract are presented, and the smart contract code is available on Github.

2. Related work

2.1. *Data trading*

Data trading has become a hot research topic in recent years. The existing representative research results are summarized as follows. Gao et al. (Gao et al., 2020) designed a differentiated private crowd-aware data trading mechanism to protect the identity privacy of consumers and the task privacy of crowd workers during the data collection process. Dai et al. (Dai et al., 2020) proposed a secure data trading ecosystem to solve the problem that data brokers and buyers can access the seller's raw data. Oh et al. (Oh et al., 2019) proposed a data trading model for data brokers. This solves the lack of transparency between data providers and data brokers/consumers. Zhao et al. (Zhao et al., 2019) proposed a new fair data trading protocol based on blockchain to verify the data availability of data users, protect the privacy of data providers and realize the fairness of payment between data providers and data consumers. Jung et al. (Jung et al., 2018) proposed a set of accountability protocols for dishonest consumers in big data trading to achieve a secure big data trading environment. Gao et al. (Gao et al., 2018) proposed an enhanced privacy protection auction scheme by using the concepts of homomorphic encryption and secure network protocol design. This solves the privacy protection problem of data auction in cyber-physical systems. Cao et al. (Cao et al., 2017) proposed an iterative auction mechanism to coordinate data trading in data markets with multiple data owners, data collectors and data users, so as to distribute large amounts of data to heterogeneous users with different interests. Delgadosegura et al. (Delgadosegura et al., 2017) proposed an innovative protocol to ensure fairness between data sellers and data buyers. Therefore, neither party needs to trust the other party during the entire trading process, because fairness is enforced by this protocol. To support data trading between data sellers and data buyers, An et al. (An et al., 2017) proposed a multi-round false name proof auction scheme to protect the best auction results from false name bidding attacks. Iyilade and Vassileva (Iyilade and Vassileva, 2013) proposed a user data sharing strategy framework based on adaptive purpose to pro-

tect user privacy and achieve personalized services required for user data sharing.

To the research of data trading, scholars have made a lot of contributions. In addition, in terms of data trading research, there is another issue worthy of research, namely, data resale. Therefore, the first data resource protection blockchain solution using smart contract is proposed to eliminate the risk of data resale.

2.2. *Blockchain*

The use of blockchain is a hot research topic and has attracted widespread attention from scholars and practitioners in recent years. Representative studies on the use of blockchain are as follows. Demirkan et al. (Demirkan et al., 2020) studied the current and potential uses of blockchain technology in business, especially in accounting and network security. Research results show that blockchain affects auditing in different ways. This will greatly change the industry. In addition, blockchain should be effectively implemented in different aspects of network security and accounting, such as auditing and general accounting procedures. Zhang et al. (Zhang and Chen, 2020) reviewed the existing research on Industry 4.0, Internet of Things, blockchain and business analytics, and provided a comprehensive overview of the latest research on these topics. The results help scholars figure out the future research directions of these topics. To identify and promote the development of blockchain technology, Lu (Lu, 2019) reviewed the existing research on blockchain and its key components, blockchain-based IOT, blockchain-based security, blockchain-based data management and main applications based on the technology. This study provides a comprehensive overview of the latest blockchains and describes a forward-looking direction. Viriyasitavat et al. (Viriyasitavat et al., 2019) explored how the blockchain technology and the Internet of Things can benefit from innovating business models. This provides a good starting point and reference point for new researchers or those interested in adopting these technologies in enterprises. Xu et al. (Xu and Viriyasitavat, 2019) proposed a smart contract to establish process execution trust suitable for the IoT environment. A consensus method to select validators extended from Practical Byzantine Fault Tolerance (PBFT) is introduced to solve the time and bias challenges. Hassani et al. (Hassani et al., 2018) summarized the opportunities and challenges from the perspective of bankers, so as to give the most comprehensive overview of the impact of blockchain in the banking industry. In addition, they also discussed the impact of big data from the blockchain on bank data analysis in the future, and showed that filtering and signal extraction are increasingly important to the banking industry. Lu (Lu, 2018) deeply analyzed the basic characteristics and categories of blockchain, and described practical applications. Through the analysis of existing applications and technologies, the development prospects of blockchain have been found.

To the research on the use of blockchain, scholars have made many contributions. This lays the foundation for us to propose a data resource protection blockchain solution based on smart contract. The proposed data resource protection blockchain solution solves the problem of data resale and effectively protects the rights and interests of data owners. This will promote the flow and sharing of data resources. In addition, the proposed blockchain solution also ensures the integrity, confidentiality, non-repudiation and anti-tampering of the entire data resource protection process, as well as the transparency and traceability of events that have occurred.

3. Definitions and models

3.1. *All participating entities*

In the data resource protection solution, there are four entities: data trading center, data owner, file server and arbitration institution. Each entity has its own responsibility in the process of data resource protection.

(1) Data Trading Center (DTC): Data trading center is the creator/owner of smart contract. It is responsible for broadcasting data-for-sale information to the nodes in the blockchain.
(2) Data Owner (DO): Data owners provide their data. They are willing to share and trade the data.
(3) Inter Planetary File System (IPFS): IPFS is a network transport protocol designed to create shared files with persistent distributed storage. It is a content addressable peer-to-peer hypermedia distribution protocol. The nodes in the IPFS network will form a distributed file system.
(4) Arbitration Institution (ABR): Arbitration institution is the entity trusted by data trading center, file server and data owners. It can verify that the terms and conditions of a smart contract are in the interest of all participating entities. Here, assume that the arbitrator will not violate the principle of fairness and justice under any circumstances.

3.2. *Adversary model & channel assumption*

Data owners are divided into two categories: malicious data owners and trusted data owners. Assume that in any data market, there are malicious data owners who resell others' data for profit, and there are trusted data owners who report malicious data owners' behavior for reward.

(1) Malicious Data Owner (DO_m): Malicious data owners avoid any trading-related liability and resell others' data. Here, assume that the malicious data owners want to resell X-data. In addition, assume that the malicious data owners resell data only through this system. It is worth noting that when the malicious data owners want to resell others' data, they will try to add a lot of imposters to perturb the data.
(2) Trusted Data Owner (DO_n): Trusted data owners sell only the data they collect/generate, and do not resell others' data. Here, assume that the data generated/collected by the trusted data owners is Y-data.
(3) Channel Assumption: Here, assume that all the participating entities communicate through secure communication channel. Out-of-chain communication is encrypted and decrypted by pre-distributed keys to ensure that data resources are not open to the public. It also means that the authorization and authentication of the identity are in

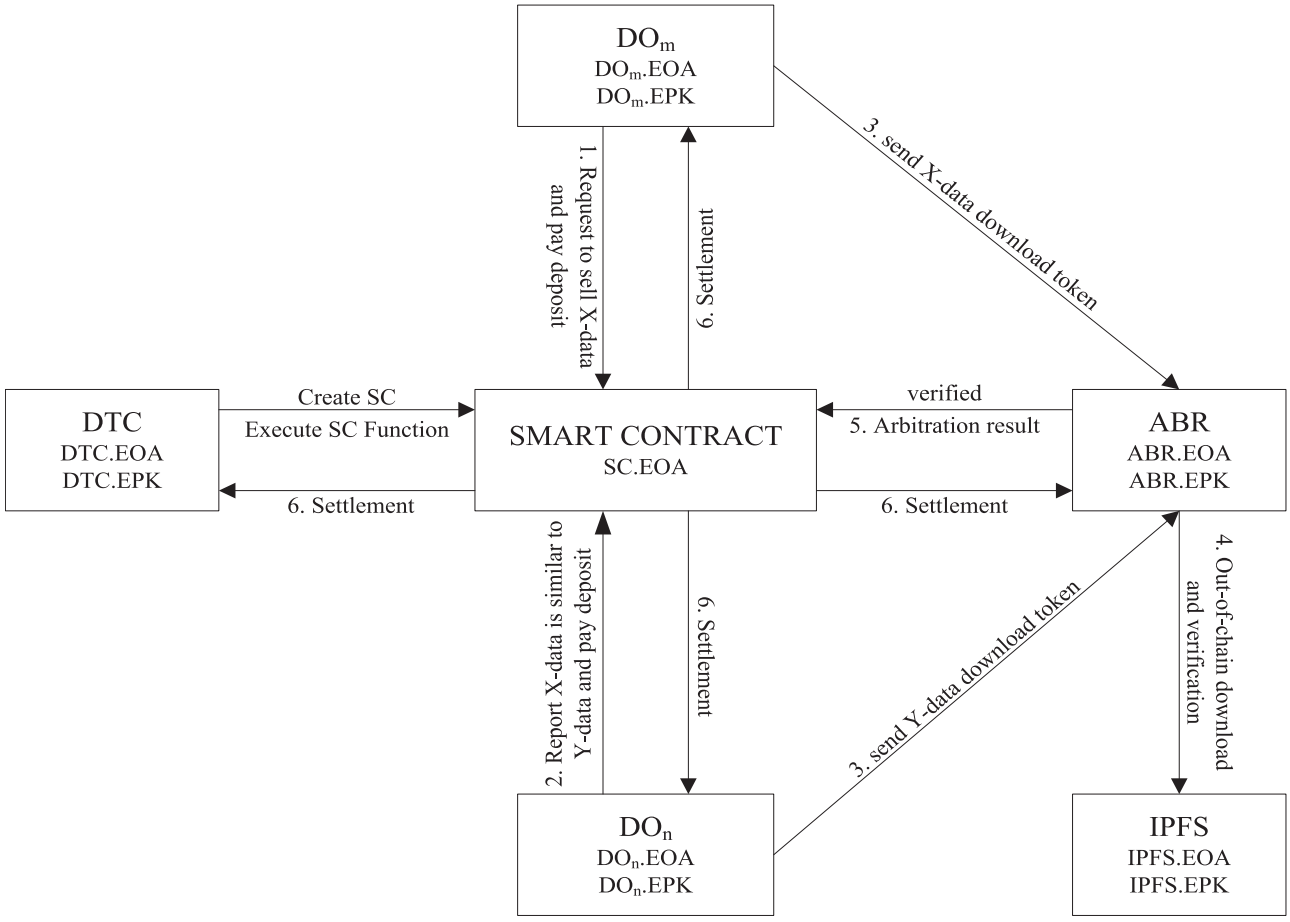


Fig. 1 – The data resource protection framework.

place, because all the participating entities need to communicate through the correct entity key.

3.3. *Data resource protection model*

The data resource protection framework is built around smart contract, as shown in Fig. 1. In Fig. 1, EOA represents the public key (that is, the Ethereum address), and EPK represents the private key. Participative entities in the framework have Ethereum addresses to ensure smooth communication between entities and smart contracts on the chain. In addition, because the content of the data-for-sale agreement requires the consent of all the participants, the terms and conditions of the agreement are stored in advance on the IPFS server. At the same time, the hash value provided by the IPFS storage file can be obtained through the IPFS server. The IPFS hash value is a key. It is used to view the complete content of the agreement in a content-addressed manner. Therefore, the IPFS hash value of the terms and conditions of the agreement will be part of the smart contract. It is worth noting that blockchain is very expensive to store large blocks of data. IPFS is a decentralized, open source point-to-point file server. It can store large blocks of data at low cost, thus blockchain is only used to store hash values provided by IPFS storage files (Benet, 2014).

The data resource protection process is summarized as follows:

If the DO_m agrees to the terms and conditions of data-for-sale agreement, he/she can send a request for "data-for-sale" to the smart contract. At the same point, he/she must pay a deposit to the smart contract. After paying the deposit, the data trading center adds the DO_m to the list of data sellers by executing the function of the smart contract. After that, it broadcasts the detailed description information of the "X-data-for-sale" to the nodes in the blockchain.

If the DO_n has no objection to the X-data through monitoring the information, the X-data will be officially sold online in the future. Then the deposit paid by the DO_m is refunded. If the DO_n has an objection to the X-data, he/she thinks that the X-data is similar to the Y-data. The DO_n deems that the DO_m may have the behavior reselling the data. Then, the DO_n sends an arbitration request to the smart contract and pays a deposit.

After the intervention of the arbitration institution, the DO_m and the DO_n will respectively generate a timeliness token and send it to the arbitration institution. The arbitration institution downloads the X-data and the Y-data from the IPFS by the tokens. Then, it measures the similarity between the X-data and the Y-data by the large-margin multi-task metric learning in an off-chain way. If the arbitration result is similar-

data, the DO_m will be judged to have the behavior reselling the data. Then, the deposit paid by the DO_m will be proportionally distributed to the DO_n, the data trading center and the arbitration institution. Meanwhile, the detailed description information of the "X-data-for-sale" will be deleted from the smart contract by executing the function of the smart contract. If the result of arbitration is data-dissimilarity, the DO_n is judged to have the behavior misreporting the DO_m. Then, the DO_n should have to pay a fine to the arbitration institution. After that, the data trading center will broadcast the detailed description information of the "X-data-for-sale" to the nodes in the blockchain.

4. Specifications of the data resource protection model

4.1. *Data-for-sale mechanism*

When the DO_m wants to sell the X-data, he/she sends a request to the smart contract. After the DO_m pays a deposit, the data trading center adds the DO_m to the list of data sellers. Then, the data trading center broadcasts the detailed description information of the "X-data-for-sale" to the nodes in the blockchain. The structure of the data-for-sale mechanism is summarized as follows:

(1) Request: The DO_m uses the Ethereum address (DO_m.EOA) to send a request for the "X-data-for-sale" to the smart contract. In the meantime, the deposit has been paid to the smart contract.
(2) Response: The data trading center responds to the request information of the DO_m and adds the DO_m to the list of data sellers by executing the function of the smart contract. Then, it broadcasts the detailed description information of the "X-data-for-sale" to the nodes in the blockchain.
(3) Dispute: The DO_n's opinion on the X-Data.
(4) Result: If the DO_n has no objection to the X-data, the X-data will be sold online in the future. Then the deposit paid by the DO_m is refunded. If the DO_n considers that the X-data is similar to the Y-data, the dispute will be transferred to the dispute resolution mechanism.

4.2. *Dispute resolution mechanism*

When the DO_n considers that the X-data is similar to the Y-data, he/she requests the arbitration institution to intervene in the dispute and pays a deposit to the smart contract. When the arbitration is conducted by the arbitration institution, the arbitration result will determine whether the X-data can be sold in the future. The structure of the dispute resolution mechanism is summarized as follows:

(1) Report: The DO_n uses the Ethereum address (DO_n.EOA) to send a report information to the smart contract. The information content is that the DO_m may have the behavior reselling the data. Meanwhile, the deposit has been paid to the smart contract.
(2) Response: The smart contract responds to the report information of the DO_n. Then it sends a message to the arbitration institution. The message content is to request the arbitration institution to intervene in the dispute. At the same time, the smart contract sends information to the DO_m and the DO_n respectively. The information content is the token needed to download the data. After that, the arbitration institution will receive the tokens needed to download the X-data and the Y-data. After the arbitration institution downloads the data, it measures the similarity between the X-data and the Y-data by the large-margin multi-task metric learning in an off-chain manner.
(3) Arbitration: The similarity between the X-data and the Y-data.
(4) Result: If the arbitration result is data-dissimilarity, the X-data will be sold on the chain. Then, the deposit paid by the DO_m is refunded. In the meantime, the DO_n must pay a fine to the arbitration institution. If the result of arbitration is similar-data, the data trading center will delete the relevant information of the DO_m by executing the function of the smart contract. After that, the deposit paid by the DO_m will be proportionally sent to the data trading center, the DO_n and the arbitration institution.

4.3. *The functions of smart contract*

The data resource protection model is built around the smart contract. The validation of the model will be achieved by writing the functional code of the smart contract. Therefore, the functions required by the smart contract are shown in Table 1.

4.4. *Out-of-chain communication*

If the arbitration institution wants to download data from the IPFS, it needs to get a token for downloading the data. The role of the token is as a credentials for downloading data. This ensures the confidentiality of data resources.

When the data-for-sale dispute occurs and the DO_n has paid a deposit, the DOmGenerateTokenm() function and the DOnGenerateTokenn() function in the smart contract will respectively generate the token for downloading X-data ($token_x$) and the token for downloading Y-data ($token_y$). The generated tokens are unique to the arbitration institution. A token is a hash created by using built-in keccak256 function of Solidity (Solidity, 2020). The Ethereum address of the arbitration institution (ABR.EOA), the Ethereum address of the DO_m (DO_m.EOA)/the Ethereum address of the DO_n (DO_n.EOA), the Ethereum address of the data trading center (DTC.EOA), the name of the X-data (DO_m.DataName)/the name of the Y-data (DO_n.DataName), the block time-stamp for the X-data (BTS_x)/the block time-stamp for the Y-data (BTS_y), and the token validity for the X-data (TV_x)/the token validity for the Y-data (TV_y) are used as the hash components to generate the $token_x$/the $token_y$. Therefore, the $token_x$ and the $token_y$ are unique to the arbitration institution, that is, the $token_x$ = keccak256 (ABR.EOA, DO_m.EOA, DTC.EOA, DO_m.DataName, BTS_x, TV_x), and the $token_y$ = keccak256 (ABR.EOA, DO_n.EOA, DTC.EOA, DO_n.DataName, BTS_y, TV_y).

The arbitration institution uses the above tokens to download the data from the IPFS. When the data is downloaded

Table 1 – The functions of smart contract.

Function	Description
RequestSellData()	DO sends a request for "data-for-sale" to smart contract.
AddDOm()	DTC adds DO_m's information to the list of data sellers.
RemoveDOm()	DTC deletes DO_m's information from smart contract.
PayReportDeposit()	DO_n pays a deposit.
DOmRefund()	DO_m applies for refund of deposit.
DOmGenerateTokenm()	X-data download token is generated by calling the function.
DOnGenerateTokenn()	Y-data download token is generated by calling the function.
DTCComfirmedResult()	DTC calls this function to determine how to handle the request for "data-for-sale".
DisputeResolution()	ABR solves the data-for-sale dispute by calling this function.
Settlement()	The deposit paid by DO_m will be confiscated by calling this function.
Penalty()	This function is used to punish the misreporting behavior of DO_n.
SCStatus()	DTC calls this function to decide whether to enable or disable smart contract.

and authenticated, the message passed between the arbitration institution and the IPFS is associated with the signature. This is done by the private key of the arbitration institution (ABR.EPK), namely ABRSigned (ABRMsgIPFS) = ABR.EPK (Hash (ABRMsgIPFS). The arbitration institution sends a message related to its signature to the IPFS. The message includes the arbitration institution's token ($ABR.token_x$)/($ABR.token_y$), the Ethereum address of the arbitration institution (ABR.EOA), the message sending time-stamp for the X-data ($ABR.TS_x$)/the message sending time-stamp for the Y-data ($ABR.TS_y$), the Internet protocol address (ABR.IP), and the Ethereum address of the IPFS (IPFS.EOA), that is, ABRMsgIPFS =$ABR.token_x$&ABR.EOA&$ABR.TS_x$&ABR.IP&IPFS.EOA/ABRMsgIPFS = $ABR.token_y$&ABR.EOA&$ABR.TS_y$&ABR.IP&IPFS.EOA.

The IPFS will use the information received through the smart contract to verify the message received from the arbitration institution. After validating the message sent by the arbitration institution, the IPFS will reply to the message. The reply message includes the Internet protocol address (IPFS.IP), the message validation time-stamp (IPFS.TS), the Ethereum address of the IPFS (IPFS.EOA), and the Ethereum address of the arbitration institution (ABR.EOA), namely IPFSMsgABR = IPFS.IP&IPFS.TS&IPFS.EOA&ABR.EOA. The reply message is signed by the private key of the IPFS and sent to the arbitration institution, namely, IPFSSigned (IPFSMsgABR) = IPFS.EPK (Hash (IPFSMsgABR). The purpose of the reply message is that the arbitration institution can verify the IPFS. After the mutual authentication is successfully completed, an open Secure Socket Layer (SSL) connection is used for the data exchange. Then the data resource is transferred from the IPFS to the arbitration institution. The process for the arbitration institution to download the X-data and the Y-data from the IPFS in an out-of-chain manner is shown in Figs. 2 and 3.

4.5. *Large-margin multi-task metric learning*

Parameswaran and Weinberger proposed large-margin multi-task metric learning through large-margin nearest neighbor classification and multi-task learning paradigm (Parameswaran and Weinberger, 2010). It uses the shared Mahalanobis metric of $M_0 \geq 0$ and the task-specific characteristics of additional matrix $M_1, \ldots M_T \geq 0$ to model the commonalities between different tasks. It defines the distance of task t as $d_t(X_i, Y_j) = \sqrt{(X_i, Y_j)^T (M_0 + M_t)(X_i - Y_j)}$ An important aspect of multi-task learning is the proper coupling of multi-task learning. It must be ensured that the learning algorithm does not overemphasize shared parameter M_0 or single parameter $M_1, \ldots, M_T$. To ensure this balance, the following regularization term is used: $\min_{M_0,\ldots,M_T} \gamma_0 \|M_0 - I\|_F^2 + \sum_{t=1}^{T} \gamma_t \|M_t\|_F^2$, where F and I represent Frobenius norm and identity matrix, respectively. The trade-off parameter γ_t controls the regularization of M_t for all t = 0, 1,... T.

If there is a data-for-sale dispute, the arbitration institution can arbitrate the similarity between the data by utilizing the large-margin multi-task metric learning in an off-chain way. This machine learning algorithm is helpful for the arbitration institution to make a correct judgment on the data-for-sale dispute.

5. Implementation of the smart contract

In this section, the implementation scheme of smart contract is proposed by designing sequence diagrams and algorithms. The implementation scheme includes three sequence diagrams and three algorithms. The full code of smart contract[1] is made available for all the details.

5.1. *Sequence diagrams*

If the DO_m agrees with the terms and conditions of data-for-sale agreement, he/she can make a request for "data-for-sale" by calling the RequestSellData() function. In the meantime, the DO_m has paid a deposit. By executing the AddDOm() function, the data trading center adds the Ethereum address of the DO_m, the relevant information of the X-data to the list of data sellers of the smart contract. After that, the data trading center broadcasts the detailed description information of the "X-data-for-sale" to the nodes in the blockchain.

The DO_n learns that there is the "X-data-for-sale" by monitoring the information. At this point, he/she will judge whether the X-data is similar to the Y-data based on the infor-

[1] https://github.com/106968687/DRP-SC.

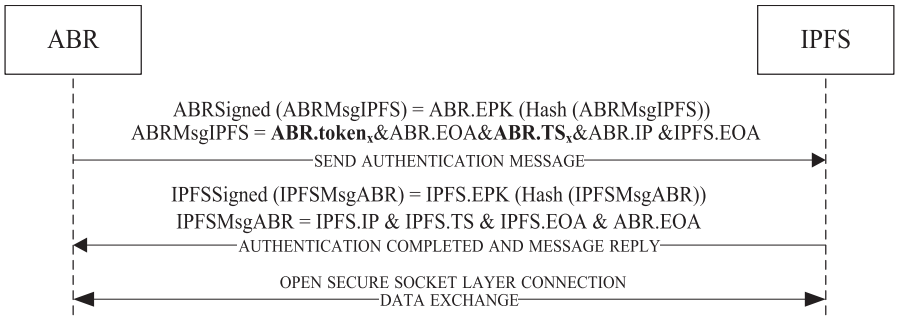


Fig. 2 – The process of downloading the X-data.

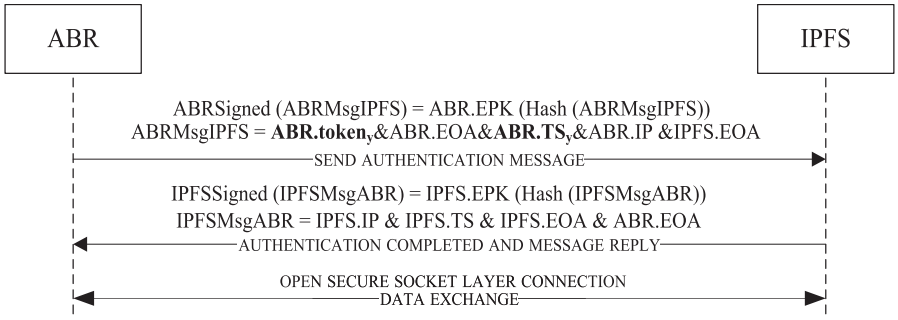


Fig. 3 – The process of downloading the Y-data.

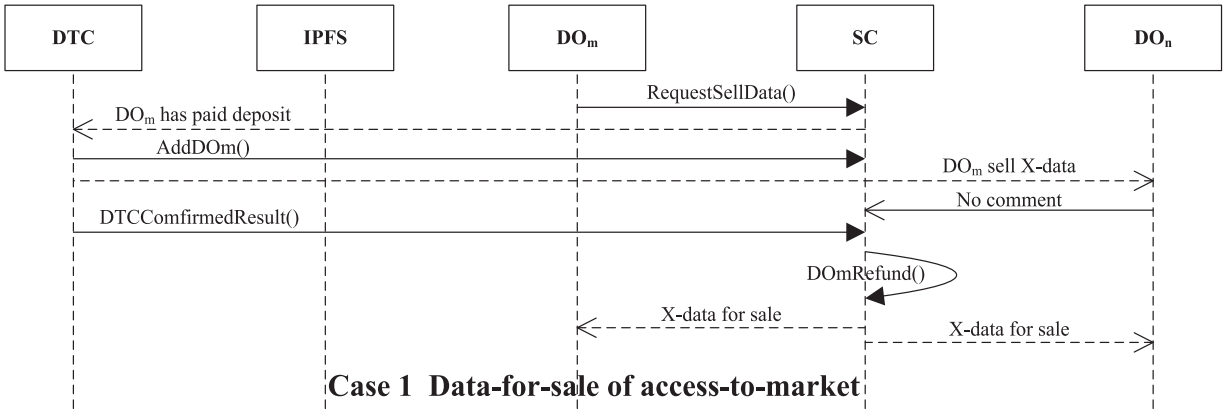


Fig. 4 – The sequence diagram of "data-for-sale" of "access-to-market".

mation obtained. The DO_n's judgment will determine whether to take action to report the DO_m's behavior reselling the data.

If the DO_n has no objection to the X-data, the X-data will be officially sold online in the future. Then the deposit paid by the DO_m will be refunded by calling the DOmRefund() function. The sequence diagram of "data-for-sale" of "access-to-market" is shown in Fig. 4.

If the DO_n has an objection to the X-data, he/she will report the DO_m's behavior reselling the data by calling the PayReportDeposit() function. Meanwhile, the deposit has been paid to the smart contract. At this time, the data trading center will broadcast the information reported by the DO_n to the nodes in the blockchain. After receiving the information, the arbitration institution will intervene in the dispute. At the same time, the DOmGenerateTokenm() function and the DOnGenerateTokenn() function will generate the tokens needed to download the X-data and the Y-data. Then the tokens will be sent to the arbitration institution. Subsequently, the arbitration institution will use the tokens received to download the X-data and the Y-data from the IPFS by the out-of-chain communication. After the arbitration institution downloads the data, it will judge the similarity between the X-data and the Y-data through the large-margin multi-task metric learning algorithm. This arbitration process is conducted in an off-chain manner.

If the result of arbitration is data-dissimilarity, the Penalty() function and the DOmRefund() function will be called to punish the misreporting behavior of the DO_n and refund the deposit paid by the DO_m. After that, the data trading center will broadcast the information of the result of arbitration to

the nodes in the blockchain. The sequence diagram of data-for-sale dispute resolution (The result of arbitration is data-dissimilarity) is shown in Fig. 5.

If the arbitration result is similar-data, the Settlement() function is called to punish the DO_m's behavior reselling the data and refund the deposit paid by the DO_n. After that, the data trading center will delete the relevant information of the DO_m by calling the RemoveDOm() function. In the meantime, it broadcasts the information of the arbitration result to the nodes in the blockchain. The sequence diagram of data-for-sale dispute resolution (The arbitration result is similar-data) is shown in Fig. 6.

5.2. *Algorithmic design*

In this section, the important algorithms used in the code will be described in a vectorized manner. The details of the algorithm implementation can be seen in the complete code of the smart contract. The address in the footnote above can obtain the complete code of the smart contract.

(1) "Data-for-sale" algorithm

The DO_m uses the IPFS hash value provided by the smart contract to view the terms and conditions of data-for-sale agreement stored on the IPFS. If the participating entities have no objection to the agreement, they can send a data-for-sale request to the smart contract. For example, the DO_m sends a request for the "X-data-for-sale" to the smart contract and pays a deposit. The algorithm of "data-for-sale" is shown in Algorithm 1.

(2) "Access to market" algorithm

After the DO_m sends a request for the "X-data-for-sale" to the smart contract and pays a deposit, the data trading center adds the Ethereum address of the DO_m and the relevant information of the X-data to the list of data sellers. Then, it broadcasts the detailed description information of the "X-data-for-sale" to the nodes in the blockchain. If the DO_n has no objection to the X-data, the X-data will be sold on the chain, namely, the X-data-for-sale of "access-to-market". After that, the deposit paid by the DO_m will be returned. The algorithm of "access to market" is shown in Algorithm 2.

Algorithm 1 – Data-for-sale.

Input: EOA, Deposit, contractStatus, DO_mStatus
Output: X-data-for-sale
1: EOA ← Ethereum address.
2: Deposit ← deposit value.
3: contractStatus ← contract status.
4: DO_mStatus ← DO_m status.
5: **if** msg.value = Deposit **then**
6: **if** contractStatus = waiting for DO_m **then**
7: DO_m.EOA.send(msg.value).
8: DO_mStatus ← DO_mPaid.
9: Create a notification that DO_m paid deposit.
10: Create a notification that X-data-for-sale.
11: **else**
12: Revert contract status and display errors.
13: **end if**
14: **else**
15: Revert contract status and display errors.
16: **end if**

Algorithm 2 – Access-to-market.

Input: EOA, DO_mStatus, DO_mpaid, ABRReceivedToken, Deposit
Output: X-data-for-sale of access-to-market
1: EOA ← Ethereum address.
2: DO_mStatus ← DO_m status.
3: DO_mpaid ← DO_mStatus value.
4: ABRReceivedToken ← DO_mStatus value.
5: Deposit ← deposit value.
6: **if** DO_mStatus == DO_mpaid or DO_mStatus == ABRReceivedToken **then**
7: **if** DO_mStatus == DO_mpaid **then**
8: Create a notification that X-data-for-sale of access-to-market.
9: DO_m.EOA.transfer(Deposit).
10: Create a notification that X-data will be sold online in the future.
11: **else**
12: **if** DO_mStatus == ABRReceivedToken **then**
13: Execute DisputeResolution().
14: Turn to Algorithm 3.
15: **else**
16: Revert contract status and display errors.
17: **end if**
18: **end if**
19: **else**
20: Revert contract status and display errors.
21: **end if**

(3) "Dispute resolution" algorithm

If the DO_n has an objection to the X-data, he/she thinks that the X-data is similar to the Y-data. The DO_n sends a request for reporting "the DO_m's behavior reselling the data" to the smart contract and pays a deposit. The data trading center will send a request for the arbitration to the arbitration institution. The arbitration institution will intervene in the dispute. The arbitration institution downloads the data from the IPFS by obtaining the $token_x$ and the $token_y$. After the data download is completed, the arbitration institution will arbitrate the similarity between the X-data and the Y-data through the large-margin multi-task metric learning. The arbitration process is completed in an off-chain way. If the result of arbitration is data-dissimilarity, the DO_n will be penalized for the misreporting behavior. If the arbitration result is similar-data, the DO_m will be punished for the behavior reselling the data. The algorithm of "dispute resolution" is shown in Algorithm 3.

6. Testing and validation of the smart contract

Solidity language will be used to write the smart contract. Remix IDE will be used for programming and debugging of the smart contract (Welcome to Remix Documentation, 2020).

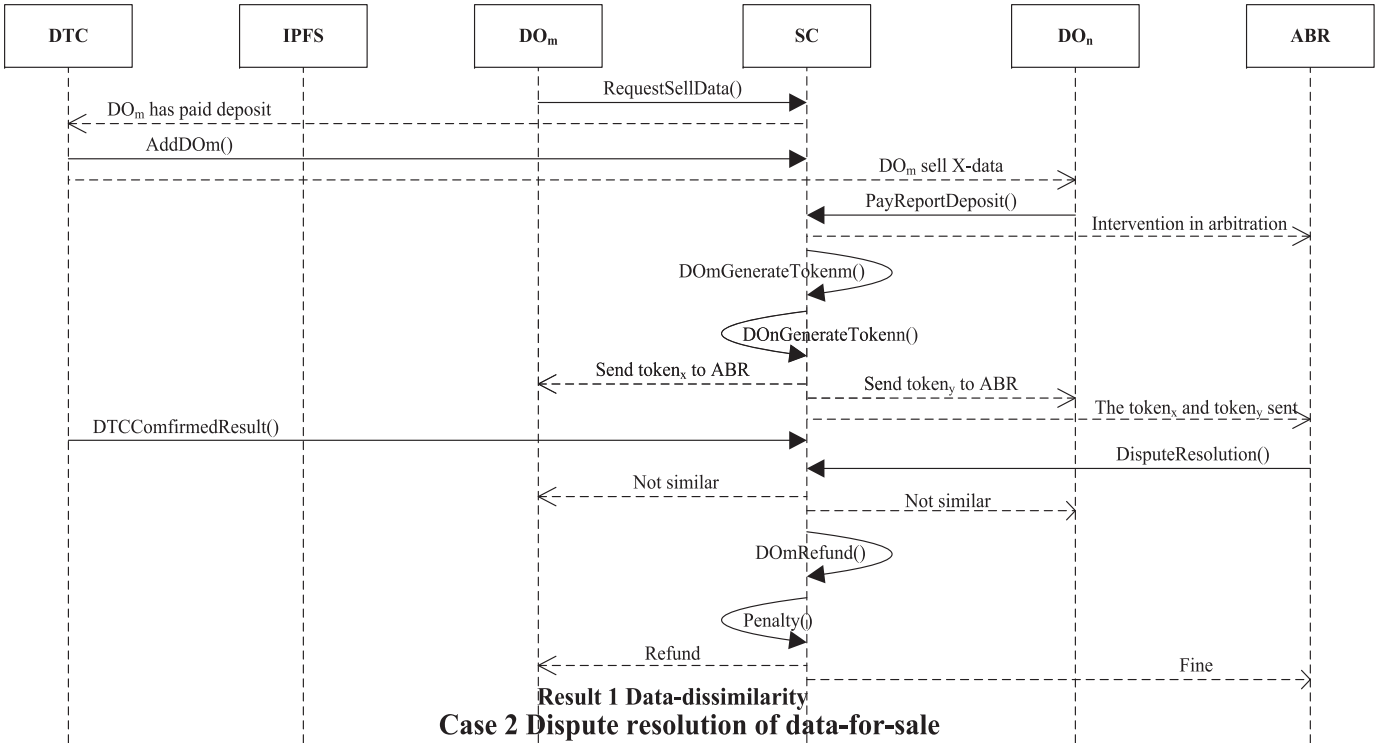


Fig. 5 – The sequence diagram of data-for-sale dispute resolution (The result of arbitration is data-dissimilarity).

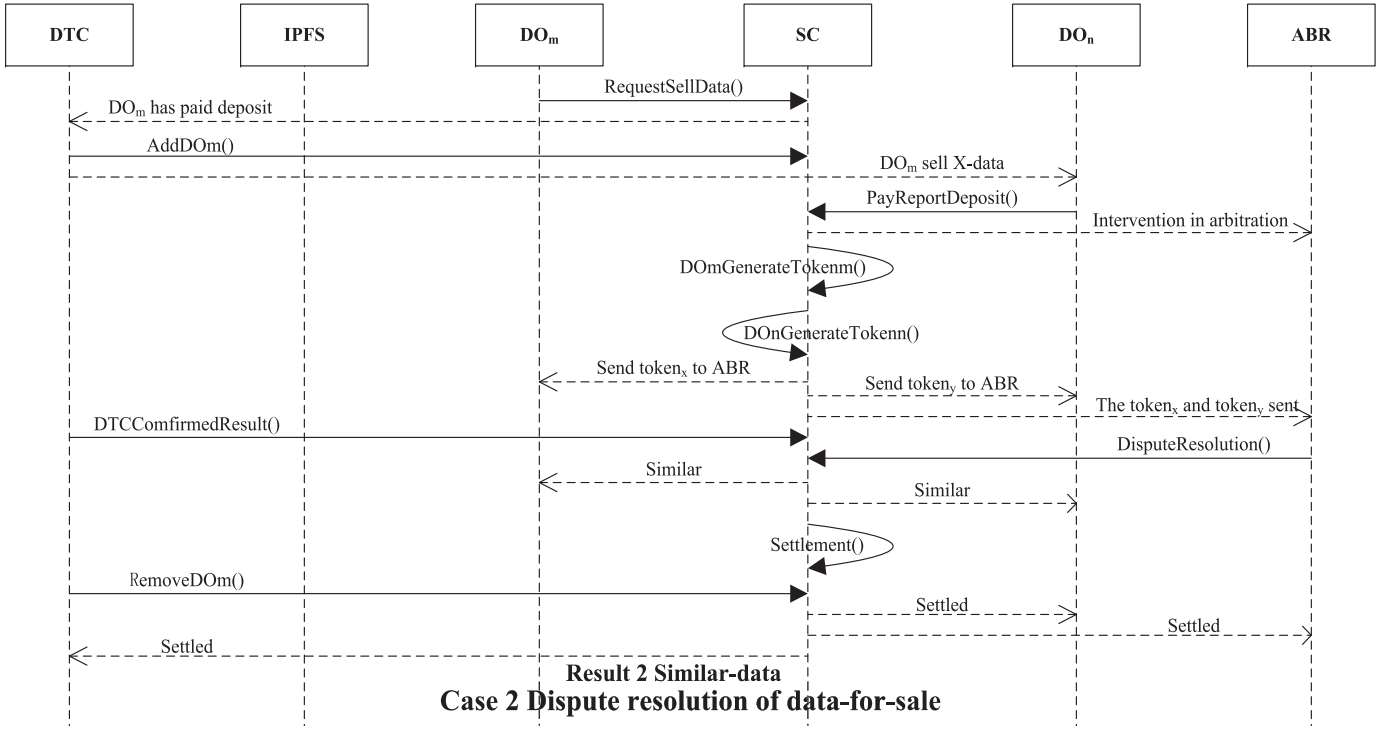


Fig. 6 – The sequence diagram of data-for-sale dispute resolution (The arbitration result is similar-data).

Algorithm 3 – Dispute resolution.

Input: EOA, Result, DO_mStatus, DisputeEnterMarket, Deposit
Output: Dispute has been resolved.
1: EOA ← Ethereum address.
2: Result ← arbitration result.
3: DO_mStatus ← DO_m status.
4: DisputeEnterMarket ← DO_mStatus value.
5: Deposit ← deposit value.
6: **if** DO_mStatus == DisputeEnterMarket **then**
7: **if** Result == true **then**
8: Result ← similar data.
9: Execute Settlement().
10: DO_n.EOA.transfer(Deposit).
11: **else**
12: Result ← data dissimilarity.
13: Execute Penalty().
14: DOm.EOA.transfer(Deposit).
15: **end if**
16: **else**
17: Revert contract status and display errors.
18: **end if**

Ethereum wallet will be used to test and validate the smart contract.

6.1. *Test preparation*

The smart contract will be tested and validated by building a private chain and using the Ethereum wallet. The main functions of the smart contract is tested and validated through the experiments. The main functions include the DO_m's X-data for sale, the DO_n reporting the X-data, access to market and dispute resolution.

The Ethereum addresses of DTC, DO_m, DO_n, IPFS and ABR are respectively "0 × 70113e01419B976c2072c07f9Ed3A47bB23E44BC", "0 × 54f0a8692dD2158F00886C653C05c1c3179D118d", "0 × 30E46cdf20931888aC86c801a4f0E81d7a465B07", "0xDc90683afff86425c2EF0D878398441D33d190E8", and "0 × 5B368D5a8FdfcFB94D336104Ff0125D92c9688D8". The functions of the smart contract are performed by the specific entities. The entities are authorized by utilizing the modifiers in the Solidity language and the uniqueness of the Ethereum address.

6.2. *Validation of the main functions*

The validation of the main functions will be presented in the form of events that have occurred, according to the characteristics of smart contract.

(1) The DO_m's X-data for sale

The DO_m uses the RequestSellData() function to send a request for the "X-data-for-sale" to the smart contract. Meanwhile, the deposit has been paid to the smart contract. At this time, the DO_m is in a paid status. After the successful execution of the function, the request for "data-for-sale" is completed. The sent request will be processed by the data trading center. Fig. 7 (a) shows the original balance of the DO_m's account. Fig. 7 (b) shows a list of functions of smart contract. Fig. 7 (c) shows the balance of the DO_m's account after the DO_m executes the RequestSellData() function. The DO_m must pay the gas to the miner as a reward for validating the executed function.

(2) The DO_n reporting the X-data

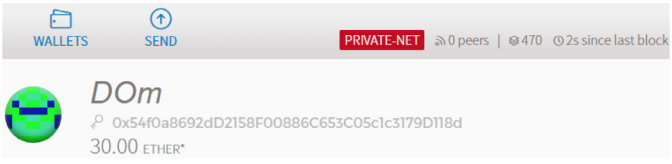
The DO_n uses the PayReportDeposit() function to send a request for reporting the X-data to the smart contract. In the meantime, the deposit has been paid to the smart contract. At this point, the DO_n is in the paid state. After the DO_n pays the deposit, the arbitration institution will intervene in the dispute. When this function is executed, a token generation event occurs. Fig. 8 (a) shows the original balance of the DO_n's account. Fig. 8 (b) shows the balance of the DO_n's account after the DO_n executes the PayReportDeposit() function. Fig. 8 (c) shows the occurrence of $token_x$ and $token_y$ generation events.

(3) Access to market

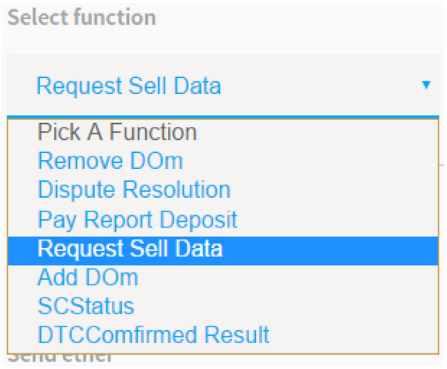
The DO_m sends a request for the "X-data-for-sale" to the smart contract and pays a deposit. If the DO_n has no objection to the X-data, the data trading center confirms the X-data-for-sale of "access-to-market" by executing the DTCComfirmedResult() function. At this time, the deposit paid by the DO_m will be returned. Fig. 9 (a) shows the account balance of the DO_m. Fig. 9 (b) shows the occurrence of both "access to market" and "the DO_m refunded" events. Fig. 9 (c) shows the account balance of the DO_m after the DTC executes the DTCComfirmedResult() function.

(4) Dispute resolution

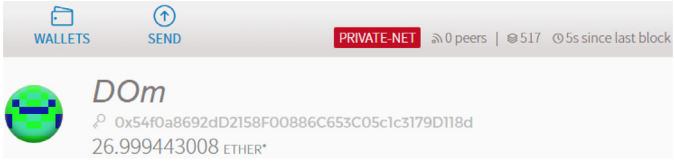
If the DO_n reports that the X-data is similar to the Y-data, the data trading center transfers to the arbitration institution for the dispute by executing the DTCComfirmedResult() function. After obtaining the $token_x$ and the $token_y$, the arbitration institution downloads the data from the IPFS in an out-of-chain communication way. After that, it uses the machine learning algorithm in an off-chain manner to judge the similarity between the X-data and the Y-data. If the X-data is not similar to the Y-data, the Penalty() function is executed and the DO_n is fined. If the X-data is similar to the Y-data, the Settlement() function is performed and the DO_m is punished. Fig. 10 (a) shows the account balances for DTC, DO_m, DO_n, and ABR. Fig. 10 (b) shows the event occurrence of the arbitration institution intervening in a dispute and conducting arbitration. Fig. 10 (c) shows the arbitration result is similar-data, and the deposit paid by DO_m is proportionally allocated to DTC, DO_n, and ABR. It is worth noting that since DTC is the creator of the smart contract and the blockchain has been mining, the number of ether owned by DTC has been increasing. Therefore, the number of ether allocated to DTC cannot be visually shown in the figure.



(a)

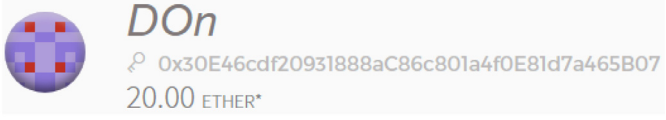


(b)

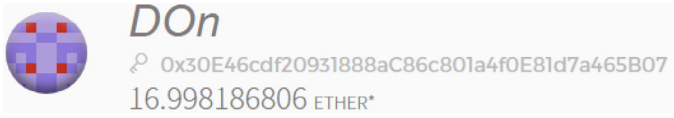


(c)

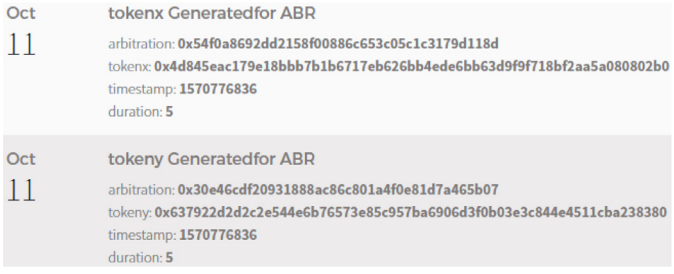
Fig. 7 – (a) The original balance of the DO_m's account. (b) A list of functions of smart contract. (c) The balance of the DO_m's account after the DO_m executes the RequestSellData() function.



(a)



(b)



(c)

Fig. 8 – (a) The original balance of the DO_n's account. (b) The balance of the DO_n's account after the DO_n executes the PayReportDeposit() function. (c) The occurrence of $token_x$ and $token_y$ generation events.

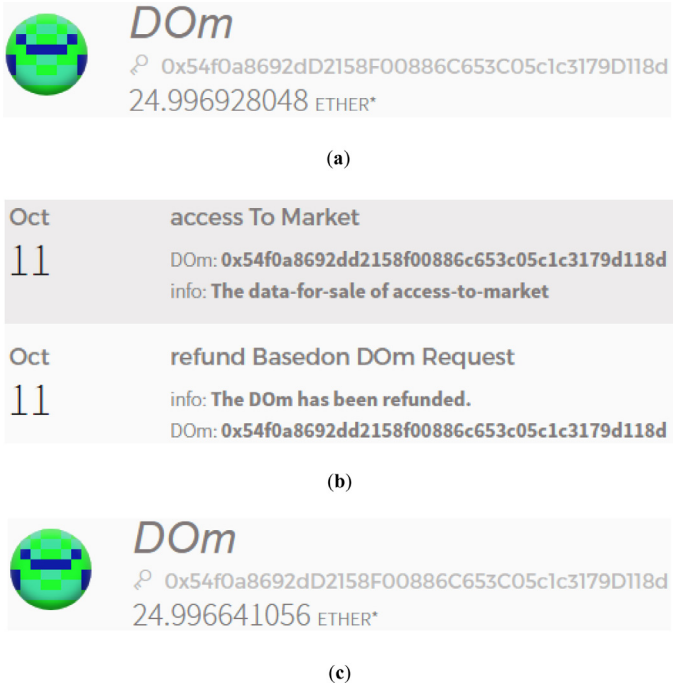


Fig. 9 – (a) The account balance of the DO_m. (b) The occurrence of both "access to market" and "the DO_m refunded" events. (c) The account balance of the DO_m after the DTC executes the DTCComfirmedResult() function.

7. Limitations and discussions

7.1. *Reasons for using smart contract*

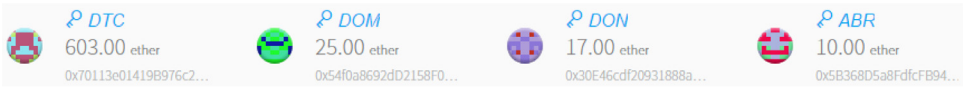
Because the centralized data trading center is the stakeholder of data trading, there is a problem of colluding with other stakeholders in data trading to seek greater profits. This is why a data resource protection solution is built around smart contract.

The advantages of using smart contract are as follows: (1) Smart contract is an automatic computer program. It not only has the functions of receiving, processing, storing and sending information, but also can control assets and respond to external received information. (2) Smart contract must be deployed in blockchain to play a role. Blockchain has the key characteristics of decentralization, tamper-proof, traceability and asymmetric encryption. Its characteristics can enable all participating nodes in the blockchain to receive data-for-sale information in time, so that the participating nodes can quickly feedback opinions on the data-for-sale.

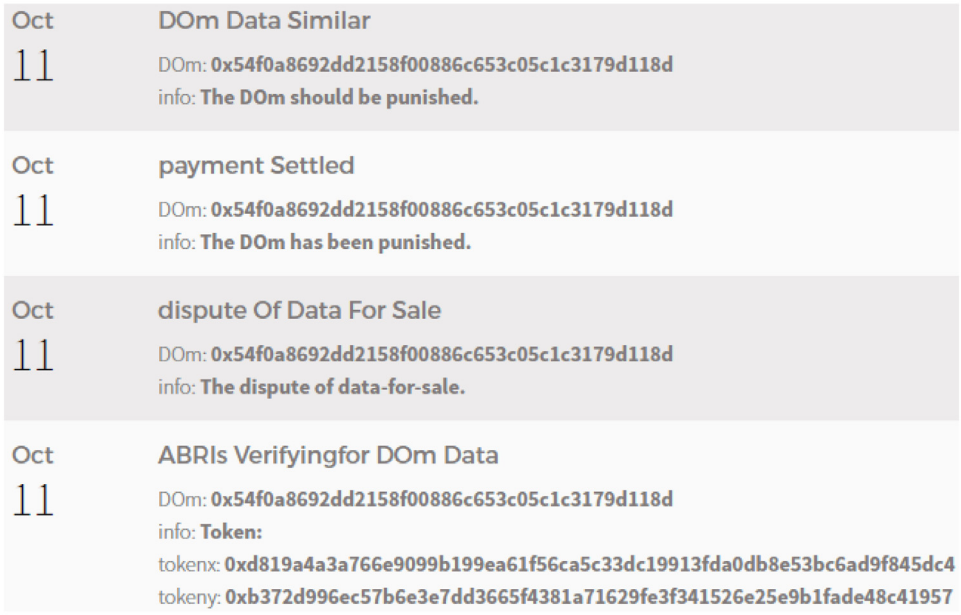
7.2. *Security characteristics*

The key security requirements and functions of the data resource protection solution are discussed through the analysis of authentication and authorization, confidentiality, non-repudiation and integrity.

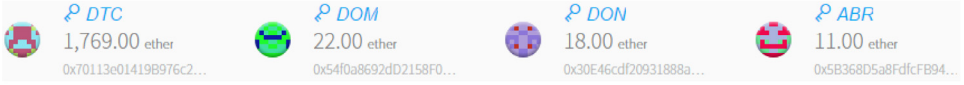
(1) Authentication and Authorization. The functions of smart contract can only be performed by the specific authorized entities. An error warning occurs if the entity calling the function is identified as unauthorized. Then all completed status will be restored. It is worth noting that the spent gas will not be returned. With regard to out-of-chain communication between the user and IPFS, this depends on a successful authentication handshake. This generates a secure SSL connection to download the data resource.
(2) Confidentiality. The confidentiality is to ensure that no one can interpret communication between any two parties or access data without authorization. Although confidentiality can be achieved by encryption, it is also necessary to maintain the confidentiality of out-of-chain communication between users and IPFS. This ensures that only authorized users can access the data stored in IPFS. In the solution, after a successful authentication handshake, the message is encrypted and decrypted through a secure SSL session. This ensures mutual authentication between users and IPFS. Using smart contract in the solution and deploying it on the blockchain can reduce the burden of using PKI for encryption, because the current PKI system is centralized, lack of transparency, and relies on Trust Certificate Authorization in key distribution. Furthermore, blockchain is used for decentralized and asymmetric encryption, and smart contract is used to receive and process information and assets. In addition, each entity has a unique Ethereum address. Ethereum addresses have asymmetric public and private key pairs. This can be used to encrypt messages transmitted over an SSL session.
(3) Non-repudiation. Communications and events on the Ethereum blockchain are part of the distributed ledger log. The originator of all operations is logged. No one can change the log. No one can deny the actions they initiated because they are recorded in the tamper-proof dis-



(a)



(b)



(c)

Fig. 10 – (a) The account balances for DTC, DO_m, DO_n, and ABR. (b) The event occurrence of the arbitration institution intervening in a dispute and conducting arbitration. (c) The arbitration result is similar-data, and the deposit paid by DO_m is proportionally allocated to DTC, DO_n, and ABR.

tributed ledger log. For out-of-chain communication, if malicious entities attempt to mimic the user's Ethereum address/unique token, their actions will be discovered through the signature information, because they do not know the correct private key for signing the message.

(4) Integrity. The advantage of adopting blockchain is that all messages exchanged between participating entities are tamper-proof. It means that no one can change the message. The out-of-chain communication between users and IPFS can be secured by using timestamps and unique user tokens. The token has been passed on the chain. The advantage of this is that the communication can prevent replay and MITM attacks.

7.3. *Advantages and disadvantages*

A data resource protection solution based on smart contract is proposed. In this solution, smart contract can handle not only information, but also assets. Once deployed in the blockchain, the smart contract cannot be changed. This eliminates the risk of human intervention in information and asset processing, which makes the solution more credible. The decentralized, traceable, tamper-proof and asymmetric encryption nature of blockchain makes the operation of all entities traceable and all unauthorized operations blocked, which makes this solution more reliable. IPFS can store large amounts of data encrypted at a low cost, which makes this solution more feasible. Large-margin multi-task metric learning can solve the problem of similarity between data, which makes the solution more convincing. In addition, the proposed dispute resolution mechanism is concise and effective. This is because the arbitration result obtained by the designed dispute resolution mechanism is an accurate conclusion, not an approximate conclusion. The dispute resolution process can directly get accurate conclusions, and this process is simple and effective.

Since this paper is the first research to propose a data resource protection solution based on blockchain, smart con-

tract, IPFS and machine learning, the maturity of this solution in practice needs to be further verified. Furthermore, although the proposed solution can fulfill the responsibility of preventing data resale behavior in this system, there is a limitation that the system cannot prevent the buyer from reselling the purchased data to the outside of the system. One possible solution to this limitation is to encrypt data usage times. Once the data usage times are exhausted, the data can no longer be used. However, this solution increases the cost of data trading, so it will cause the corresponding data sales price to increase. In addition, the arbitrator may have the problem of being compromised, thus making a ruling that violates the principle of fairness and justice. A possible solution to this limitation is to establish a penalty system within the arbitration institution. If the arbitrator violates the arbitration principle, he/she will be punished in accordance with the internal punishment mechanism of the arbitration institution, so as to ensure that the arbitrator resolves the dispute in accordance with fair and just arbitration rules. If an arbitrator violates the principle of fairness and justice, it will not only damage the credit of the arbitration institution, but also damage the interests of all data trading participants.

8. Conclusion

In this paper, a data resource protection solution is proposed to solve the data resale problem. Firstly, the data resource protection solution from "data-for-sale request", "access to market" to "dispute resolution" is realized through the construction of data-for-sale and dispute resolution mechanisms. Next, the data-for-sale of "access-to-market" are controlled, penalty measures are set and disputes are resolved through the design of "data-for-sale", "access to market" and "dispute resolution" algorithms. Furthermore, the smart contract code is programmed and debugged by the Remix IDE. The test and verification of the smart contract is realized through the private chain and Ethereum wallet. Finally, the security characteristics of the solution are discussed, such as confidentiality and non-repudiation.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Wei Xiong: Conceptualization, Methodology, Software, Validation, Writing - original draft. **Li Xiong:** Writing - review & editing, Supervision, Project administration.

REFERENCES

AlTawy R, ElSheikh M, Youssef AM, Gong G. In: Cryptol. ePrint Arch. Lelantos: a blockchain-based anonymous physical delivery system; 2017. Tech. Rep. 2017/465.

An D, Yang Q, Yu W, et al. Towards truthful auction for big data trading. In: 2017 IEEE 36th International Performance Computing and Communications Conference (IPCCC). IEEE; 2017. p. 1–7.

Azaria A, Ekblaw A, Vieira T, Lippman A. MedRec: using blockchain for medical data access and permission management. Proc. Int. Conf. Open Big Data (OBD) 2016(Aug.):25–30.

Benet, J. Ipfs-content addressed, versioned, p2p file system," 2014. Available Online: https://arxiv.org/abs/ 1407.3561 (Feb. 1, 2020).

Cao X, Chen Y, Liu KJR. Data trading with multiple owners, collectors, and users: an iterative auction mechanism. IEEE Trans. Signal Inf. Process. Over Netw. 2017;3:268–81.

Cruz JP, Kaji Y, Yanai N. RBAC-SC: role-based access control using smart contract. IEEE Access 2018;6:12240–51.

Dai W, Dai C, Choo KR, et al. SDTE: a secure blockchain-based data trading ecosystem. IEEE Trans. Inf. Forensics Security 2020:725–37.

Delgadosegura S, Perezsola C, Navarroarribas G, et al. In: A Fair Protocol for Data Trading Based on Bitcoin Transactions. Future Generation Computer Systems; 2017. p. 832–40.

Demirkan S, Demirkan I, McKee A. Blockchain technology in the future of business cyber security and accounting. J. Manag. Anal. 2020 online published.. doi:10.1080/23270012.2020.

Gao W, Yu W, Liang F, et al. Privacy-preserving auction for big data trading using homomorphic encryption. IEEE Trans. Netw. Sci. Eng. 2018.

Gao G, Xiao M, Wu J, et al. DPDT: a differentially private crowd-sensed data trading mechanism. IEEE Internet of Things J. 2020;7(1):751–62.

Hasan HR, Salah K. Proof of delivery of digital assets using blockchain and smart contracts. IEEE Access 2018;6:65439–48.

Hassani H, Huang X, Silva E. Banking with blockchain-ed big data. J. Manag. Anal. 2018;5(4):256–75.

He HW, Yan A, Chen ZH. Survey of smart contract technology and application based on blockchain. J. Comput. Res. Dev. 2018;55:2452–66.

IPFS is the Distributed Web. Available Online: https://ipfs.io/ (Feb. 1, 2020).

Iyilade J, Vassileva J. A framework for privacy-aware user data trading. In: International Conference on User Modeling, Adaptation, and Personalization. Berlin, Heidelberg: Springer; 2013. p. 310–17.

Jung T, Li XY, Huang W, et al. AccountTrade: accountability against dishonest big data buyers and sellers. IEEE Trans. Inf. Forensics Secur. 2018;14:223–34.

Lu Y. Blockchain: a survey on functions, applications and open issues. J. Ind. Integr. Manag. 2018.

Lu Y. The blockchain: State-of-the-art and research challenges. J. Ind. Inf. Integr. 2019:80–90.

Meng W, Tischhauser EW, Wang Q, Wang Y, Han J. When intrusion detection meets blockchain technology: a review. IEEE Access 2018;6:10179–88.

Nakamoto, S. Bitcoin: a peer-to-peer electronic cash system. Available Online: https://bitcoin.org/bitcoin.pdfN (Feb. 1, 2020).

Oh H, Park S, Lee GM, et al. Personal data trading scheme for data brokers in IoT data marketplaces. IEEE Access 2019:40120–32.

Parameswaran S, Weinberger KQ. Large margin multi-task metric learning. Adv. Neural Inf. Process. Syst. 2010:1867–75.

Solidity. An object-oriented, high-level language for implementing smart contracts. Available Online: https://solidity.readthedocs.io/en/latest/ (Feb. 1, 2020).

Sun Y, Song H, Jara AJ, Bie R. Internet of Things and big data analytics for smart and connected communities. IEEE Access 2016;4(Mar):766–73.

Szabo, N. Formalizing and securing relationships on public networks. Available Online: http://www.firstmonday.org/ojs/index.php/fm/article/view/548/469 (Feb. 1, 2020).

Tian F. In: A supply chain traceability system for food safety based on HACCP, blockchain & internet of things; 2017. p. 1–6.
Treleaven P, Brown RG, Yang D. Blockchain technology in finance. Computer 2017(9):14–17.
Viriyasitavat W, Anuphaptrirong T, Hoonsopon D. When blockchain meets internet of things: characteristics, challenges, and business opportunities. J. Ind. Inf. Integr. 2019:21–8.
Welcome to Remix Documentation. Available Online: https://remix.readthedocs.io/en/latest/ (Feb. 1, 2020).
Xiong W, Xiong L. Smart contract based data trading mode using blockchain and machine learning. IEEE Access 2019;7:102331–44.
Xu L, Viriyasitavat W. Application of blockchain in collaborative Internet-of-Things services. IEEE Trans. Comput. Soc. Syst. 2019;vol. 6(6):1295–305.
Yang MJ. A design of data trading platform based on cryptology and blockchain technology. Inf. Commun. Technol. 2016;10:24–31.
Zhang C, Chen Y. A review of research relevant to the emerging industry trends: industry 4.0, IoT, block chain, and business analytics. J. Ind. Integr. Manag. 2020;5(1):165–80.
Zhao YQ, Yu Y, Li YN, et al. Machine learning based privacy-preserving fair data trading in big data market. Inf. Sci. 2019:449–60.

Wei Xiong is currently pursuing the Ph.D. degree with the Department of Information Management, School of Management, Shanghai University. His research interests include security, smart contract, blockchain, and information systems.

Li Xiong is currently a full-time Professor with the Department of Information Management, School of Management, Shanghai University. Her main research interests include information systems, security, and collaborative innovation.