An Oil and Gas Big Data Sharing Model Based on Blockchain Technology

Yuyan Wang1,*, Shujun Huang1 and Xiang Yu2

1Strategic Research Center of Oil and Gas Resources, MNR, Beijing, China
2Beijing Shida Zhengxin Mineral Resources Technology Consulting Co., Ltd., Beijing, China

*Corresponding author: wangyy@sinooilgas.org.cn

Abstract. With the continuous in-depth application of new technologies such as big data and Internet of Things in the oil and gas field, the scale and value of oil and gas data continue to rise, laying the foundation for oil and gas big data sharing. Oil and gas big data sharing can effectively integrate different data sources and break the dilemma of "data islands" where data in the oil and gas field are separated from each other. However, there are problems in actual practice such as (1) difficulties in trust among different government departments, scientific research institutions and enterprises, (2) insufficient data privacy protection, and (3) weak data traceability. In response to the above problems, this article proposes a model of oil and gas big data sharing based on blockchain technology. We build a blockchain for oil and gas big data sharing alliances to secure oil and gas big data by utilizing the decentralization and non-tampering characteristics. We also provide supports for oil and gas big data sharing by controlling data access strictly based on the smart contracts. The evaluation and analysis results show that this model makes oil and gas big data more secure and authoritative compared with the traditional centralized sharing method.

1. Introduction

With the continuous deepening of informatization in the oil and gas field, and the popularization and application of technologies such as big data, Internet of Things, and artificial intelligence, the value of oil and gas big data is getting higher and higher. However, these data are often stored by different government departments, scientific research institutions, and enterprises, making the entire industry trapped in the dilemma of "data islands". The contradiction between data security and data sharing has become more and more prominent, which has seriously affected the sharing and opening of oil and gas big data [1], and has restricted the innovation and development of the oil and gas industry. Oil and gas big data sharing can effectively tap the potential value of data and realize a thorough transformation of the strategy. This helps to provide strong supports for awakening a large number of unused data assets, avoiding repeated construction, and building an interconnected data sharing system.

Traditional oil and gas data sharing often adopts a centralized sharing model, where the data is concentrated and stored on the central server of an authoritative intermediary, and the authoritative intermediary conducts unified sharing and publishing [2]. This traditional sharing model has many security problems such as malicious tampering of centralized data, lack of transparency leading to
untrustworthiness, loss of data ownership by data owners, single point of failure, and difficulty in data traceability.

In view of the above analysis, this paper designs an oil and gas big data sharing model based on the untamperable, decentralized, and traceable characteristics of blockchain technology [3]. It provides a guarantee for the safe sharing of oil and gas big data, and greatly improves the authority and reliability of oil and gas data. The efficiency can be further improved in many fields such as exploration and development, oil and gas storage and transportation, oil and gas trading, strategic planning, and situation analysis. Therefore, accelerating the research on blockchain-based oil and gas big data sharing not only has theoretical research significance, but also has practical application value.

2. Background

2.1. Hash algorithm and Merkle tree

Hash algorithm is a function that can map data of different sizes to fixed-length strings. The hash value of the data can be easily calculated, but it is difficult to calculate the original data from the hash value. The Merkle tree is a tree structure based on a hash algorithm. Each non-leaf node is the hash value of its leaf node [4]. In the blockchain system, the Merkle tree is used to verify whether the data has been tampered with. We calculate the hash value representing for the whole data set in the Merkle tree, and generates a unique Merkle root, which is stored in the block header. This strategy helps to reduce the computational workload, by avoiding re-calculating the hash values of all data blocks whenever they are used.

As shown in Figure 1, the data file is divided into small data blocks, and the hash value of each data block is calculated separately as the bottom leaf node of the Merkle tree. In the bottom-up calculation process, we do not calculate the hash values of all data blocks, but concatenating the left and right adjacent hash values to calculate a new hash value for this block. If the number of hash values is odd, the remaining last hash value is directly hashed. The process is performed recursively, until the last root
hash is Merkle Root. Assuming that data A in the figure changes, you only need to recalculate Hash 0-0 and Hash 0 to get the Merkle Root.

2.2. Consensus mechanisms
Decentralization is one of the core advantages of blockchain technology. Unlike a centralized system, blockchain technology uses a consensus mechanism to coordinate the consistency between nodes in a distributed system [5] to ensure that all nodes participate in the competition fairly. Account rights, to prevent nodes from colluding to deceive to destroy the credibility of the system, and provide a solution to the problems of high cost and low efficiency in centralized systems. The main consensus mechanisms currently include: Proof of Work (PoW), Proof of Stake (PoS), Proof of Share Authorization (DPoS), and Practical Byzantine Fault Tolerance (PBFT).

PoW consumes computing power to perform a certain amount of difficult calculations, and obtains a verified hash result, and uses a certain amount of computing power to avoid service abuse and malicious attacks [6]. Malicious nodes are difficult to possess enough computing power to tamper with data is currently widely used in public chains (for example: Bitcoin). PoS applies the game theory method by considering that the greater the benefits of the nodes, the more they hope that the system will operate safely and fairly. According to the equity mechanism to allocate the accounting rights, the node has a higher probability to obtain the accounting rights if it possesses a higher benefit from the system. This strategy avoids the waste of resources caused by the meaningless hash calculations of most nodes in the PoW consensus mechanism. However, this mechanism is prone to unfairness, and nodes with large stakes are vulnerable to network attacks. As a very efficient consensus mechanism, the key of DPoS is to select the consensus nodes by voting. However, the entire consensus process still relies on the requirements for trust. Therefore, the degree of DPoS decentralization is not high and extremely relies on the integrity of consensus nodes. PBFT reduces the time complexity of the original Byzantine Fault Tolerant (BFT) algorithm to a polynomial level to improve the efficiency. In addition, PBFT is based on calculations and does not require rewarding mechanism. Therefore, the consensus mechanism of PBFT is more suitable to construct the oil and gas big data alliance chain.

2.3. Proxy re-encryption
The concept of proxy re-encryption was proposed by Balze et al. [7] at the 1998 European Cryptography Annual Conference. At the 2005 Network and Distributed System Security Symposium (NDSS), Ateniese et al. [8] provides a standardized formal definition of proxy re-encryption, focusing on its application in distributed storage. Proxy re-encryption is mainly to solve the inconvenience of users when sharing data, to ensure that each participant cannot obtain any plaintext information during the sharing process. This strategy realizes the conversion between ciphertexts, and enhances the security and reliability of data sharing.

The proxy re-encryption process involves three roles: data producer, data consumer, and agent. When the data owner wants to share encrypted data to the data consumer, the data owner will generate the proxy re-encryption key for the data consumer and transmit the proxy key to the agent. According to the proxy key, the agent uses the proxy re-encryption algorithm to re-encrypt the encrypted data and send it to the data consumer. The data consumer uses his private key to decrypt the re-encrypted data to the original plaintext data to finish the data sharing process.

2.4. Smart contract
Smart contract is a computer protocol embedded in hardware or software that can be executed automatically, allowing trusted transactions without a trusted third party. It was first proposed by Nick Szabo in 1995 [9], but due to lack of the trusty operating environment, smart contracts have not been applied to actual business scenarios.
Figure 2. Flow chart of smart contract operation.

The birth of blockchain technology provides a credible execution platform for smart contracts. It saves the instructions and running status of the smart contract in the block, and automatically triggers preset instructions through real-time monitoring to form an irreversible execution process [10]. The execution process can be tracked in order to ensure the efficiency, transparency and credibility of the entire process. Smart contracts are created and called in the form of transactions in the blockchain, and the contract instructions are executed on distributed nodes. Each node verifies each other and agrees on the execution result. There is no central node, so any node failure will not affect the operation of smart contracts.

Ethereum is specifically designed for smart contracts and is currently the largest blockchain supporting smart contracts [11]. It is an open source platform for building and publishing smart contracts. As an extension of blockchain technology, it can be executed in Ethereum Virtual Machine based on the Turing-complete language, where developers can use high-level languages such as Solidity and C++ to develop and publish smart contracts. This greatly improves the usability of smart contracts and provides a guarantee for developing countless applications.

3. The sharing model for the big data of oil and gas

3.1. Global design of the model
As shown in Figure 3, the oil and gas big data security sharing model based on blockchain technology is mainly composed of five parts: data producers, data consumers, smart contracts, oil and gas big data sharing alliance blockchain, and data sources:

1) Data producers. It refers to government departments, scientific research institutions and enterprises that have data ownership. They are mainly responsible for publishing data abstracts, setting permissions for sharing content, and managing and storing encrypted source data.

2) Data consumers. It refers to government departments, scientific research institutions, and enterprises that have data demands. They can only access the data within the authorized scope when authorized by the data producer.

3) Smart contract. It is responsible for interactive operations such as deploying the data summary on the chain, setting data sharing permissions, data summary query, and data permission verification. These operations will be recorded in the blockchain to ensure the traceability of the data sharing process.

4) Oil and gas big data sharing alliance block chain. It is mainly composed of government departments, scientific research institutions, and enterprises with sufficient data processing capabilities to jointly maintain the alliance chain and share various oil and gas data. The scope of oil and gas big data mainly includes the transfer of mineral rights, registration of mineral rights, reserves review, mineral reserves, mineral development, exploration and mining supervision, strategic research, planning
research, resource evaluation, situation analysis, policies and regulations, standards and specifications, information resources. Each node publishes and shares according to its own needs. Only the data summary is stored on the chain, and the specific data is still in the node's own data source.

5) Data source. The data producer encrypts the shared data and stores it in the data source to ensure that the data source can be accessed remotely, including local databases, cloud databases and other data sources.

Figure 3. The sharing model for the big data of oil and gas.

3.2. Data sharing authority control
In order to strengthen the security and reliability of data sharing process, our model adopts a proxy re-encryption model that supports keyword search. [12] This realizes the sharing authority control of oil and gas data, improves the decryption efficiency, and makes it convenient for data demanders to query and search. Different from the traditional models, there is no definite re-encryption agent role in this model based on the decentralized characteristics of the blockchain. Any node in the alliance chain can complete the re-encryption operation, and all shared process data are recorded in the chain. This strategy ensures that the data sharing process is open, transparent and traceable.

In this model, the data producer can convert the shared ciphertext into a proxy ciphertext based on the public key of the data consumer, and the corresponding plaintext information remains unchanged. The data consumer can use the private key to decrypt the proxy ciphertext to complete the data sharing. Throughout the sharing process, smart contracts are used to interact, in order to ensure that the sharing process is safe and reliable. The whole process includes 11 steps including (1) initialization, (2) private key generation, (3) search key generation, (4) data encryption, (5) re-encryption key generation, (6) re-encryption, (7) index generation, (8) token generation, (9) re-encryption verification, (10) data decryption and (11) decryption for the re-encrypted data. The algorithms of the key steps are mentioned in details as follows:

(2) Private Key generation
Input GP as the public parameter, K as the master key, $S \subseteq U$ as the set of data producer attributes:
3rd International Conference on Green Energy and Sustainable Development
IOP Conf. Series: Earth and Environmental Science 651 (2021) 032105
doi:10.1088/1755-1315/651/3/032105

\[
\{ SK_{\sigma} = \{ K = g^a g^{at}, L = g^f, \{ K_x = H_3(x)^f \}_{x \in S} \} \} \tag{1}
\]

(6) Re-encryption
Input \( rk \) as the re-encryption key and the origin ciphertext \( CT \):

\[
CT' = \langle A_1, A_2, A_3, A_4, rk, (B_i, C_i)_{i=1}^l, D, S, (M, \rho) \rangle \tag{2}
\]

(7) Index generation
Input \( GP \) as the public parameter and the set of key words \( \text{KW} = \{ \text{kw}_i \}_{i=1}^l \)

\[
IX = \langle t_i F(k_i, t_i), IX' = \langle t_i, F(k_i, t_i) \rangle \rangle \tag{3}
\]

(11) Decryption for the re-encrypted data
Input the re-encrypted ciphertext \( CT' \), the set of attributes \( S' \) and the corresponding decryption key \( SK' \):

\[
\begin{align*}
&\left\{ e \left( A_2, H_6(A_1, A_2, (B_i, C_i)_{i=1}^l, S', (M', \rho')) \right) = e(g, D') \right\} \\
&Z' = e(A_4, \chi)_{Q_{CT}^W}, \delta \parallel \beta' = H_2(Z') \oplus A_1
\end{align*}
\]

\[
\tag{4}
\]

3.3 Smart contract system
Smart contracts are coded and deployed in the oil and gas big data alliance chain according to the demands of interaction, and are jointly managed and operated by all nodes. Transaction management and automatic business processing are realized without relying on any third-party organization. Our model uses smart contracts as an interactive method, using its decentralized, untamperable, programmable, and traceable features to realize various interactive operations and execution logic in oil and gas data sharing. It mainly includes three parts: the user identity management contract, the oil and gas data management contract, and the data authority control contract.

As shown in Table 1, the implementation of the user identity management contract includes operations such as user registration, user verification, user query, and user logout. The realization of the oil and gas data management contract includes operations such as data registration, list query, data application, and data acquisition. The data authority control contract is based on the data sharing authority to realize fine authority control of data sets, including operations such as granting sharing authority, revoking sharing authority, and authority verification.

| Contract Type                      | Contract Name          | Contract Identification |
|------------------------------------|------------------------|-------------------------|
| user identity management contract  | user registration      | userRegister            |
|                                   | user verification       | userVerification        |
|                                   | user query              | userQuery               |
|                                   | user logout             | userLogout              |
| oil and gas data management contract | data registration     | dataRegister            |
|                                   | list query              | dataList                |
|                                   | data application        | dataApply               |
|                                   | data acquisition        | dataGet                 |
| data authority control contract    | granting sharing authority | grantAuth              |
|                                   | revoking sharing authority | revokeAuth             |
|                                   | authority verification   | checkAuth               |

Table 1. The detailed list of the smart contract system.

3.4 PBFT consensus mechanism
The oil and gas big data sharing alliance blockchain first forms a master node by votes, and other nodes are slave nodes. The master node is responsible for initiating the consensus process and does not have the advantage of voting. When the master node fails, the slave nodes re-elect a new master node. Our model uses Practical Byzantine Fault Tolerance (PBFT) to achieve block consensus [13]. The detailed process is as follows:

1) Client $c$ initiates a request message to the master node, and the format is defined as $(\text{REQUEST}, o, t, c)$, here REQUEST includes the request message content $m$ and the summary information $d$ of $m$. And $o$ represents the specific operation requested, $t$ represents the request timestamp, and $c$ is the client identifier.

2) After the master node receives the client request, it will assign a sequence number to the request and generate a pre-prepared message. The format is defined as $(\text{PRE-PREPARE}, v, n, d, m)$, where $v$ is the view number, $n$ is the message sequence number, and $d$ is the summary of the message content $m$, which is then broadcast to other slave nodes for inspection.

3) After the slave node receives the pre-preparation message broadcast by the master node, it verifies whether the message $m$ has been tampered with and the validity of the sequence number $n$. After confirming that it is correct, the preparation message is generated and sent to all slave nodes. The format of the preparation message is $(\text{PREPARE}, v, n, d, i)$, where $i$ represents the number of the slave node.

4) The slave node repeats step 3 to verify the preparation message. After the verification is correct, the confirmation message is sent to other nodes through the whole network broadcast, so as to realize the common confirmation among the nodes. The format of the confirmation message is $(\text{COMMIT}, v, n, d, i)$.

5) After all nodes have confirmed the message, they will generate a result message and return it to the client. The format of the result message is $(\text{REPLY}, v, n, d, r)$, where $r$ represents the result message.

Figure 4. Blockchain consensus process of the oil and gas big data sharing alliance.

4. Evaluation and analysis

4.1. Security analysis
In this model, an asymmetric encryption algorithm is used to encrypt data. Without the private key of the data producer, even if the encrypted data is leaked, it will not lead to the disclosure of the plaintext of the data, ensuring data security. The system is based on the blockchain P2P network structure without a centralized server, and it does not rely on trusty third-party institutions. In other words, the system is completely maintained by the nodes themselves, which effectively avoids single-point attacks and ensures system stability. In the process of data sharing, proxy re-encryption technology is used to avoid the leakage of the original data in each sharing step, in order to ensure the security of data sharing.
4.2. Efficiency analysis
At present, three kinds of consensus mechanisms based on the blockchain technology (PBFT, PoW and DPoS) are widely applied in the field of oil and gas data sharing. In order to benchmark the efficiency of the three consensus mechanisms, the CPU usage is tested through experimental simulations. The experiment is deployed on a stand-alone PC (Intel(R) Core(TM) i7-4700MQ 2.40GHz CPU, 8GB memory, and Ubuntu 16.04.10 LTS Desktop). As shown in Figure 5, compared with the PoW and DPoS algorithms, the PBFT consensus mechanism used in this model has the significantly smaller CPU usage percentage, leading to the advantage in fast response.

![Figure 5. Comparison of CPU usage percentage of each consensus mechanism.]

5. Conclusions
The data sharing is of great significance to all aspects of the oil and gas field. Based on the decentralized characteristics of blockchain technology, we propose an oil and gas big data sharing model to realize data sharing among various departments, research institutions and enterprises. Compared with the traditional centralized sharing model, the security, authority and credibility have all been greatly improved.

The benchmark result of efficiency in this article generated by PC cannot fully reflect that in the actual network. In the future, we will apply virtual machines or other technologies to establish a more comprehensive experimental environment for more in-depth research.

References
[1] Feng D, Zhang M, Li H. Big Data Security and Privacy Protection [J]. Chinese Journal of Computers, 2014, 37(1): 246 – 258.
[2] Yuan Y, Wang F. Blockchain: the state of the art and future trends [J]. Acta Automatica Sinica, 2016, 42(4): 481 - 494.
[3] He P, Yu G, Zhang Y, Bao Y. Survey on Blockchain Technology and Its Application Prospect [J]. Computer Science, 2017, 44(4): 1 - 7.
[4] Merkle R C. A Digital Signature Based on a Conventional Encryption Function [C]// Conference on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1987.
[5] Yuan Y, Wang F. Blockchain: the state of the art and future trends [J]. Acta Automatica Sinica, 2016, 42(4): 481-494.
[6] Liu X. Research on blockchain performance improvement based on Byzantine fault tolerance consensus algorithm based on dynamic authorization [D]. Hangzhou: Zhejiang University, 017.
[7] BLAZE M, BLEUMER G, STRAUSS M. Invertible protocols and atomic proxy cryptography [C]
[8] Ateniese G, Fu K, Green M, et al. Improved proxy re-encryption schemes with applications to secure distributed storage[J]. M Transactions on Information and System Security (TISSEC), 2006, 9(1):1-30.

[9] WU L, MENG K, XU S, et al. Democratic centralism: A hybrid blockchain architecture and its applications in energy internet[C]/2017 IEEE International Conference on Energy Internet (ICEI). Beijing, China: IEEE, 2017:176 -181.

[10] LIND J, NAOR O, EyAL I, et al. Teetchain: Reducing storage costs on the blockchain with offline payment channels[C]/11th ACM International Systems and Storage Conference. Haifa, Israel: ACM, 2018: 125.

[11] Chen T, Zhu Y, Li Z, et al. Understanding Ethereum via Graph Analysis[C]/IEEE INFOCOM 2018 - IEEE Conference on Computer Communications. IEEE, 2018:1484-1492.

[12] LIU ZH, ZHOU P, DUAN SH, Attribute-based Proxy Re-encryption Scheme with Keyword Search[J]. Journal of Electronics & Information Technology, 2018, 40(3):684-686.

[13] Castro M, Liskov B. Practical byzantine fault tolerance and proactive recovery[J]. ACM Transactions on Computer Systems, 2002, 20(4):398-461.