International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2022): 7.942

Proposal of Blockchain based New Framework for Multi Clouds

Minjun Park

Saint Johnsbury Academy Jeju
10, Global edu - ro 304, Daejeong - eup, Seogwipo_si, Jeju Special Slf - Governing Province, Korea
Email: *s202452[at]sjajeju.kr*

Abstract: ***The emerging of cloud data sharing can create great values, especially in multi clouds. But, data island between different cloud service providers has drawn trust problem in data sharing, causing contradictions with the increasing sharing need of cloud data users. And how to ensure the data value for both data owner and data user before sharing, is another challenge limiting massive data sharing in the multi clouds. To solve the problems above, I propose a new framework with blockchain to support trustworthy and valuable data sharing. I design namespace - based unique identifier pair to support data description corresponding with data in multi - cloud, and build a blockchain - based data encoding protocol to manage the metadata with identifier pair in the blockchain ledger. To share data in multi - cloud, I build a data parsing protocol with smart contract to query and get the sharing cloud data efficiently. I also build identifier updating protocol to satisfy the dynamicity of data, and data check protocol to ensure the validity of data.***

Keywords: blockchain, multi cloud, cloud data sharing, data identifier

1. Introduction

Recently, Cloud storage is widely used because it provides users with the convenience of large - scale data storage and sharing. As a successful profit model, more and more cloud service providers are emerging. However, cloud service providers generally do not support users to obtain data resources from other cloud service providers within a single cloud platform, which leads to the emergence of "data island" in the multi clouds, and also brings inconvenience to users who want to make use of data storing in different clouds. On the other hand, with the advent of the era of big data, the value of data has continued to increase, and a paid sharing model has emerged for data sharing, in which, data owners hope that their shared data resources in the cloud can be seen by more data users. In a word, data owners and data users want to widely publish / obtain shared data resources, but cloud service providers cannot meet their needs because of "data island". This brings a big problem to data sharing in multicloud environments. In order to meet the needs of users for data sharing in multi clouds, a cloud data sharing framework based on search engine is proposed [7, 8, 9]. The URI of the data owner's shared data in the cloud is obtained by the open search engine [10, 11] through the web crawler [12, 13, 14], and provided to the data users for retrieval. Data users locate and share data in the cloud through DNS [15] resolution to realize data sharing. This framework relies on the support of cloud service providers. But in the real environments, too many crawlers have led to a decline of cloud data service quality, and there is "data island" in the multi clouds, so many cloud service providers use anti - crawler technology, which leads to the failure of multi clouds data sharing based on search engine. Another cloud data sharing framework based on centralized server is proposed [16, 17]. The shared data stored by data owners on different cloud platforms are collected and sorted into a unified format, stored in a centralized server and provided to data users. When data users want to share data, they can retrieve the unified format data to locate and obtain data. This framework is very efficient, but it faces the problem of single point failure. The failure of centralized server will cause data and economic losses to users. When the data owner needs to share his own data, he submits a registration transaction in the blockchain network, and then the blockchain encodes a unique identifier for the data to refer to the data, and records the URI of the data. Data users obtain the URI of the data by retrieving and parsing the identifier of the data in the blockchain, and finally use the URI to address the shared data in the cloud. However, the existing methods still have shortcomings in the uniqueness of the identifier and the efficiency of identifier parsing. This paper makes the following contributions: 1) I make definitions of the namespace and design the identifier pairs, which give enough consideration to both uniqueness of identifier and dynamicity of data.2) I design efficient protocols based on the identifier pairs which reduce response time of new framework and help data users to check the validity of data. In the remainder of this paper, Section 2 introduces related work, including the security and privacy protection when sharing data, and different frameworks of sharing data. The design of new framework is elaborated in Section 3. Conclusions are drawn in Section 4.

2. Related Works

There are two types of related technologies for data sharing. One is concerned with the security and privacy protection when sharing data, and the other is concerned with the frameworks of sharing data. In the following, I will introduce them separately.

2.1 Security and privacy protection when sharing data

A dynamic attribute - based access control scheme is proposed, which sets a fine - grained valid time period for each attribute and reduces the waiting time of CSP caused by manual operations. A lightweight proxy re - encryption scheme is proposed, which builds the pre - encryption algorithm and designs a certificate less protocol that remove bilinear pair and have very high performance. Qin et al. [18] introduced Shamir secret sharing scheme and permissioned

blockchain to eliminate the single point failure, and computed tokens cross domains to reduce communication and computation overhead on the data user side. Qin also made other work to realize access control based on different scenarios. There are some other related studies paying attention to internet of things and putting emphasis on privacy and security [19, 20].

2.2 Frameworks of sharing data

Different frameworks are proposed for different scenarios, such as frameworks based on SE and DNS, based on Centralized Server, and based on Blockchain.

2.2.1 Frameworks based on SE and DNS

The most famous and widely used framework based on SE and DNS is [10], in which google is proposed. In [10], Brin et al. provided the first detailed public description of large - scale Web search engine. Google crawls and indexes webs efficiently and then make heavy use of use of the structure present in hypertext to produce much more satisfying search results. People can search webs by simply typing some words or a sentence, and then google returns indexes of webs as results to the user. People can get access to the webs by clicking the indexes. However, due to the fact that so many cloud service providers use anti - crawler technology, frameworks based on search engine may not work on cloud data sharing.

2.2.2 Frameworks based on Centralized Server

Focusing on data sharing of a certain field, frameworks based on centralized server are proposed. In [16], QuerioCity, a platform to catalog, index and query highly heterogenous information coming from complex systems, is proposed. The data currency of QuerioCity is a dataset consisting of metadata, and thus, data from different sources can be used. Besides, an approach based on Semantic Web technologies is proposed to deal with the incremental and continuous integration of static and streaming data. Data from complex systems are collected and presented as indexes in the centralized QuerioCity platform. Users can get data through these indexes. However, there are always safety risks such as single point failure accompanying with centralized servers.

2.2.3 Frameworks based on Blockchain

In order to break through the limitation of shared data format, based on Namecoin and Bitcoin, Muneeb and Jude of Princeton University proposed a new system named Blockstack. It designs a system consists of several separate layers, and constructs a decentralized Internet by implementing a new public key infrastructure (PKI, non - blockchain based PKI systems can be learned through Keybase [2] and CONIKS [6]), which makes the identifier encoding and parsing system no longer limited by the data format. In order to improve the readability of the identifier, Blockstack adopts "name + namespace" as form of identifier. Ethereum [4] and others also use similar methods to encode readable identifiers. However, identifiers encoded in this way can't keep unique, so when users want to get a unique identifier, they have to try different names again and again, and finally end up hurting the readability of identifiers. To improve the uniqueness and autonomy of the identifiers, PPK, an open organization of Beijing University of Posts and telecommunications, has open source a new system named open data index name (ODIN) [1] based on Bitcoin network. ODIN introduces the height of block where registration transactions recorded and serial number of registration transaction in block to form the prefix to ensure the uniqueness of the identifier. The user's naming of the data is used as the identifier suffix to improve the readability of the identifier. The logic positions of transactions used in identifiers help ODIN to keep identifiers unique, but a big problem occurs when users update their data as a logic position cannot represents multiple transactions. In ODIN system, it is difficult to manage the update operation of ODIN records, so a web server is introduced to facilitate users to obtain ODIN. Though the data stored in the web server can be verified using information from Bitcoin, the web server still has some other risks like single point failure. Besides the problems discussed above, there are two more problems with these existing methods. The first, due to the limitation of Bitcoin, the write performance is very poor and the throughput is very low. The second, they pay little attention to validity of data besides a single signature verification, and that can never meet the need of users in big data era.

3. Design of new framework

To overcome the shortcomings of existing methods mentioned above, I propose a new framework. New framework is based on consortium blockchain, which is much more efficient than public chains like Bitcoin. New framework is designed with short - long identifier pairs, and it pays a lot of attention to validity of data. Identifiers are stored and shared with blockchain, and data can be stored in any storage entity. Here in this paper, I use cloud platforms as storage entities.

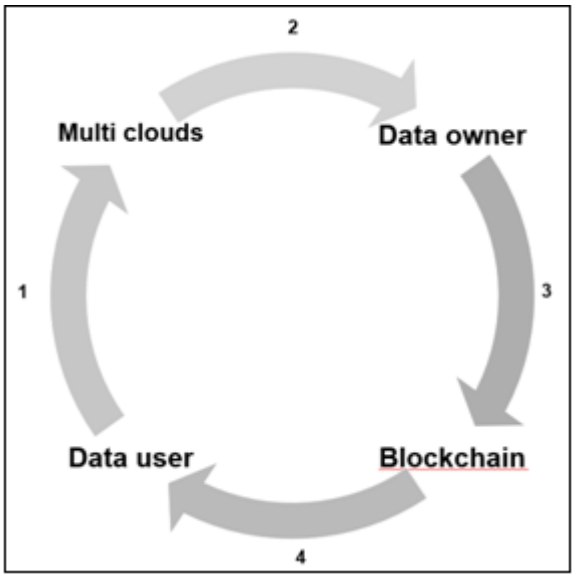

Figure 1: Overview of new framework

*** Remark**

1) Download data, Request to get data using URL
2) Upload data, Request to updata using URL, Store data and return URL, execute the request and return URL
3) Request to register date, Request to updata long identifier, Encode identifier pair and return the short identifier, Execute the request

4) Request to query data using short identifier, Request query data owner's information using short identifier, parse identifier get URL and return, Get information using the prefix of short identifier and return.

3.1 Overview

The system architecture of new framework is shown in Figure 1, which includes data owners, data users, multi clouds and blockchain network. The data owner uploads the data to the cloud and registers it in the blockchain network, who has the rights of modifying and deleting the data; the data user refers to the user who uses the data, and can query and obtain the data shared by others through the short identifier; the multi clouds stores data, which can be shared after being registered on the blockchain network; the blockchain stores metadata and runs the encoding and parsing process of data identifier. The design of identifier pairs, the encoding protocol, the parsing protocol, identifier updating protocol and data check protocol will be detailed below. Many existing methods such as ODIN and Blockstack chose Bitcoin as their underlying blockchain (although Blockstack is said that it can be applied above any blockchain, it has to make a huge change migrating to blockchains with smart contract in fact), but an obvious problem is that Bitcoin is too inefficient. The efficiency of Bitcoin cannot meet the need of a large data sharing system, and there are studies trying to solve it like Permacoin [5]. Besides, it is said there are selfish mining attacks [11] occur to Bitcoin. Here in our study, I introduce consortium blockchains as underlying blockchain. Compared to Bitcoin, consortium chains have greater throughput and shorter response time, and meanwhile its security is guaranteed. Here in this paper, I choose Hyperledger Fabric as our underlying blockchain. One of the big differences between our proposed system and the existing methods is the location of metadata. Our system stores metadata in blockchain, and I make use of the functions of the blockchain and smart contract to achieve our system's functions. Meanwhile in the existing methods such as ODIN and Blockstack, the blockchains are used as a communication channel for announcing state changes, and the metadata are stored in external entities. Regardless of the safety of external entities, they also bring another problem that the new nodes have to get a complete copy of metadata before they can join the systems, and it may lead to safety and efficiency issues.

3.2 Definitions of namespace

To ensure the uniqueness of identifiers, I make definitions of the namespace. From the perspective of users, I design "Global Domain" and "User Domain". Global Domain is the complete scope of multi clouds and blockchain, and its basic unit is the single user. Each user corresponds to a User Domain. User Domain is the data storage scope of a single user on multicloud, and its basic unit is data. The prefix and suffix of the short identifier mentioned below will be unique in the "Global Domain" and "User Domain" respectively, so as to ensure the uniqueness of the complete short identifier.

3.3 Identifier Pairs

To make new framework efficient, I designed short - long identifier pairs. The short identifiers are used to represent data, and the long identifiers are used to describe data. The long identifiers change when data is updated, and meanwhile the short identifiers keep unchanged. A short identifier is the only identification of data and it means the short identifier shall be unique. In consortium chains, I have unique certifications for each user. I name the hash of certifications cert_hash, and choose the cert_hash as the prefix of short identifier, which is unique in Global Domain, and data_name named by data owner as the suffix. The data owner shall name his data with different names to make them unique in User Domain (no need to keep different with other users' data), and then the short identifier can keep unique in the whole namespace. Beside uniqueness, the prefix actually represents a user, and it means that I can store user information, such as certification and public key, using the prefix as key, which is helpful in the following protocols. And the suffix helps data owners to have their data marked as they want. I organize all metadata into a long identifier, thus long identifier is also a tool of metadata management. Data owners can easily manage their data by managing metadata in the long identifier, and users can easily obtain information by getting metadata in long identifier. What's more, new framework is a fundamental system which have the potential to exploit different functions to adapt to different environments, the long identifier makes that possible. The advantages of identifier pairs are: 1) It is both unique and human - readable.2) It satisfies the dynamicity of data.3) The prefix of short identifier is related to owner's identity, and it avoids a lot of security issues such as counterfeiting.

3.4 Identifier encoding protocol

To share data, data owner has to register his data in new framework. I will design the identifier encoding protocol, which is also the process of data registration. Step one, data owners upload their data to clouds by sending an upload request Req_upload (data); Step two, the clouds receive and store these data, and return the URL of data to the data owners by sending a response Res_upload (URL); Step three, when data owners decide to share their data with others, they send a registration request Req_register (URL, user_sig, data_name) to blockchain; Step four, the blockchain generates a short identifier and a long identifier, and then returns the short identifier to the data owners by sending a response Res_register (short_identifier). The user_sig in the step three is data owner's digital signature of data, and the data_name is a name of data given by the data owner. The data_name should not be same with the data owner's other data_names (sequence numbers will be added at the end of data_name by new framework if and only if the data_names are same). In the step four, blockchain abstracts data owners' certification and then get hash of it as cert_hash. The cert_hash and the data_name make up a short identifier. After short identifier is generated, URL, transaction_hash, user_sig will make up a long identifier, in which the transaction_hash is generated by blockchain to represent the registration transaction. What's more, the long identifier is a tool to manage metadata, so if any metadata is needed in specific scenarios, it can be added into long identifier. A short identifier and a long identifier make up a key - value pair, and I can get metadata from long identifier using short

identifier as key. As a comparison, to register an identifier in Blockstack, users need to try again and again to see if the identifier they submit is unique over all the identifiers, and ODIN denies users the right to name their identifiers. The identifier encoding protocol I design focus on the advantages and abandons the disadvantages of Blockstack and ODIN.

3.5 Identifier parsing protocol

The short identifiers are delivered to users and the long identifiers are invisible to users. I design the identifier parsing protocol for data users to query data with short identifiers. Step one, data users query data by sending a query request Req_query (short_identifier) to blockchain; Step two, the blockchain executes the parsing process after receiving the request, and returns the metadata URL to the data users by sending a response Res_query (URL); Step three, data users send a query request Req_query (URL) to clouds to get data; Step four, clouds send the data to the data users with a response Res_query (data). In the step two, the parsing process works like this: blockchain gets long identifiers from key - value pairs with the short identifiers as keys, and then gets the needed metadata from long identifiers. Here in this protocol the needed metadata is URL. Thus, data users can easily obtain data just by a single short identifier, and meanwhile, the parsing process provides a convenient way to implement expanded functions which need to get a certain metadata. As mentioned above, Blockstack and ODIN store metadata in external entities, and thus their identifier parsing process runs outside the blockchain which may be easily attacked.

3.6 Identifier updating protocol

Data is not always static, and the content and description of data may change. I design the identifier updating protocol for data owners to update their data in both clouds and blockchain. Step one, data owners update their data in clouds by sending an updating request Req_update (data), in which the parameter data is the new version of data. Step two, the clouds receive and update these data, and return the URL of data to the data owners by sending a response Res_update (URL), in which the parameter URL may be a new one; Step three, the data owners also need to update the description of their data in blockchain, so they send an updating request Req_update (URL, user_sig) to blockchain, in which the user_sig is the digital signature of the updated data; Step four, the blockchain regenerates the long identifier after an authentication, and then sends a response Res_update (OK) to the data owners. In the step four, regenerating the long identifier means: blockchain gets long identifiers from key - value pairs with the short identifiers as keys, updates the metadata URL and user_sig in long identifiers, and then stores the key - value pairs with new values. As mentioned above, a data owner's information, such as certification and public key, is stored in blockchain and the prefix of the short identifier is the key. Thus, I can easily get the data owner's certification and do authentication. For the same reason with identifier parsing protocol, the identifier updating protocol I design is safer than existing methods. What's more, a node in Blockstack or ODIN need to read all blocks of Bitcoin to track the newest state, and meanwhile in our proposed system, the newest state is stored in blockchain which means even a new node can track the newest state without reading all the blocks or getting a copy of metadata from other nodes.

3.7 Data check protocol

After downloading the data completely, checking data is a natural idea. I design the data check protocol for data users to see a) whether the data matches the short identifier; b) whether the data is actually provided by the data owner I thought to be. Step one, data users send a data check request Req_check (short_identifier) to blockchain; Step two, the blockchain gets the metadata user_sig and user_pubkey after receiving the request, and returns the metadata to the data users by sending a response Res_check (user_sig, user_pubkey), in which the parameter user_pubkey is the data owner's public key; Step three, data users use the user_sig and user_pubkey to verify the hash of data. In step two, user_sig can be obtained from the long identifier, and user_pubkey can also be obtained from key - value pair using the prefix of short identifier as key. If the verification in step three succeeds, I know that the data actually matches the short identifier as the user_sig is obtained with the short identifier, and the data is actually provided by the data owner I thought to be as the data owner's user_pubkey is used in the verification.

4. Conclusion

Cloud storage has been widely used nowadays, but the emergence of "data island" disturbs data sharing between clouds. Protocols of previous methods cannot perform perfectly in multi clouds. In this paper, I propose our own identifier encoding and parsing system new framework. I will make definitions of namespace and design identifier pairs which achieve the goals of being readable and easy - use, and I will design protocols based on the identifier pairs.

References

[1] Wang Jiye, Gao Lingchao, Dong Aiqiang, Guo Shaoyong, Chen Hui, Wei Xin, "Block Chain Based Data Security Sharing Network Architecture Research, " Journal of Computer Research and Development, 54 (4), 742 - 749, 2017. Article (CrossRef Link)
[2] Keybase. Website (CrossRef Link)
[3] M. S. Melara, A. Blankstein, J. Bonneau, E. W. Felten, and M. J. Freedman, "CONIKS: bringing key transparency to end users, " in Proc. of 24th USENIX Security Symposium (USENIX Security 15), pp.383–398, 2015
[4] V. Buterin, "A next - generation smart contract and decentralized application platform, " white paper, 1 - 36, 2014.
[5] A. Miller, A. Juels, E. Shi, B. Parno, and J. Katz, "Permacoin: Repurposing bitcoin work for data preservation, " in Proc. of Security and Privacy (SP), 2014 IEEE Symposium on, pp.475–490, 2014. Article (CrossRef Link)
[6] I. Eyal and E. G. Sirer, "Majority is not enough: Bitcoin mining is vulnerable, " CoRR, abs/1311.0243, 2013.
[7] Pundt, Hardy, and Yaser Bishr, "Domain ontologies for

data sharing–an example from environmental monitoring using field GIS, ” Computers & Geosciences, 28.1, 95 - 102, 2002. Article (CrossRef Link)

[8] King, Gary, “An introduction to the dataverse network as an infrastructure for data sharing, ” Sociological Methods & Research, 36.2, 173 - 199, 2007. Article (CrossRef Link)

[9] Grundy, Quinn, et al., “Data sharing practices of medicines related apps and the mobile ecosystem: traffic, content, and network analysis, ” bmj, 364, 2019. Article (CrossRef Link)

[10] Brin, Sergey, and Lawrence Page, “The anatomy of a large - scale hypertextual web search engine, ” Computer networks and ISDN systems, 30.1 - 7, 107 - 117, 1998. Article (CrossRef Link)

[11] Chen, Zhen - Lin, et al., “A high - speed search engine pLink 2 with systematic evaluation for proteome - scale identification of cross - linked peptides, ” Nature communications, 10.1, 1 - 12, 2019. Article (CrossRef Link)

[12] Heydon, Allan, and Marc Najork, “Mercator: A scalable, extensible web crawler, ” World Wide Web, 2.4, 219 - 229, 1999. Article (CrossRef Link) KSII TRANSACTIONS ON INTERNET AND INFORMATION SYSTEMS VOL.15, NO.8, August 2021

[13] Thelwall, Mike, “A web crawler design for data mining, ” Journal of Information Science, 27.5, 319 - 325, 2001. Article (CrossRef Link)

[14] Farag, Mohamed MG, Sunshin Lee, and Edward A. Fox, “Focused crawler for events, ” International Journal on Digital Libraries, 19.1, 3 - 19, 2018. Article (CrossRef Link)

[15] Mockapetris, Paul, and Kevin J. Dunlap, “Development of the domain name system, ” ACM SIGCOMM Computer Communication Review, 18.4, 123 - 133, 1988. Article (CrossRef Link)

[16] Lopez, Vanessa, et al., “Queriocity: A linked data platform for urban information management, ” in Proc. of International Semantic Web Conference, Springer, Berlin, Heidelberg, pp.148 - 163, 2012. Article (CrossRef Link)

[17] Park, Kyounghyun, Minh Chau Nguyen, and Heesun Won, “Web - based collaborative big data analytics on big data as a service platform, ” in Proc. of 2015 17th international conference on advanced communication technology (icact), IEEE, 2015. Article (CrossRef Link)

[18] Qin, Xuanmei, et al., “A Blockchain - based access control scheme with multiple attribute authorities for secure cloud data sharing, ” Journal of Systems Architecture, 112, 101854, 2020. Article (CrossRef Link)

[19] Le Nguyen, Bao, et al., “Privacy preserving blockchain technique to achieve secure and reliable sharing of IoT data, ” CMC - COMPUTERS MATERIALS & CONTINUA, 65.1, 87 - 107, 2020. Article (CrossRef Link)

[20] Yang, Zhen, et al., “Protecting personal sensitive data security in the cloud with blockchain, ” Advances in Computers, 120, 195 - 231, 2021. Article (CrossRef Link)