Dynamic User Revocation in Blockchain based Framework for Data Sharing

Anil V. Deorankar[1], Pratik S. Nichit[2]

[1]*Computer Science and Engineering, Government College of Engineering, Amravati (MH), India.*
[2]*Computer Science and Engineering, Government College of Engineering, Amravati (MH), India.*

ABSTRACT

Attribute-based encryption (ABE) is an important scheme to overcome the problem of fine-grained access control and data privacy. But the private key generator PKG in ABE can decrypt the data stored in the CSP, it may bring problems such as privacy data leakage and key abuse. In addition, the traditional cloud model has a centralized storage system, thus it inherit the risk of single point of failure. With the advent of blockchain technology, decentralized storage model has come in the public view. The decentralized storage model has overcome the risk of single point of failure. It has numerous benefits over centralized storage, such as high throughput and low price. Recently Shangping Wang, Yinglong Zhang and Yaling Zhang proposed a Blockchain based framework for data sharing with fine grained access control in decentralized storage systems. However this scheme does not implement the functions of user's attribute revocation and access policy update. In order to implement user's attribute revocation functionality we combine ABE scheme, Blockchain technology and decentralized storage with DURKR model

Keywords: ABE, access control,attribute revocation, Ethereum blockchain, IPFS, smart contract.

1. INTRODUCTION

The cloud storage technology has brought great convenience to our lives making users possible to access Internet and share data at anytime and anywhere. Such cloud storage systems has achieved increasing acceptance throughout the world. However, such systems depend on large organization having strong storage and transmitting data capacity. These large organizations are regarded as a trusted third party and automatically has a drawback of single point of failure. Additionally cloud storage services providers has a risk of force Majeure (such as political censorship) lead to even an authorized users cannot access its own data. Furthermore, with the development in storage technology, the prices of storage devices are becoming smaller. The price of centralized cloud storage services mainly depends on employee wages, data center rentals and maintenance costs etc. These determined costs are fixed or gradually increases. Thus, centralized cloud storage services are costly. Decentralized storage acts as better alternative to centralized cloud storage systems. It does not make us to trust these third parties whether they will honestly store and transmit our data. Also it makes us need not to worry about our data being inaccessible. Fortunately, with the arrival of blockchain technology enabled us to implement decentralization in storage systems. It enabled us to connect peer to peer cryptocurrency [1] with storage space, CPU power, bandwidth, etc. Using decentralized storage system [2] we can rent free extra hard disk space and get returns in cryptocurrency. Users don't have to worry about they won't be able to have access to their own data, because availability of data at any time is guaranteed by the smart contracts deployed on the blockchain, and user only have to pay fees regularly for their stored data.

In traditional cloud storage systems, if a user wants to shares data secretly stored in third-party cloud server, a technology is needed to achieve fine grain access control over data that can be accessed and decrypted by only desired user. In need of this demand, the attribute-based encryption (ABE) mechanism [3] [4] was proposed and developed rapidly. In this scheme, the data owner (DO) can specifies the access policies based on the user's attributes to achieve fine grained access control over data. Almost all ABE schemes require a trusted private key generator (PKG) to setup the system and distribute all corresponding secret key to users [5]. There are oceans of problems with such system. Firstly, it is difficult to get a trusted PKG in reality. Secondly, such system faces the problem of key abuse (the control of users data is not in their own hands). The PKG has the power to decrypt all the data stored in the server, and it may leak users data for illegal gains, political censorship or other certain interests. Once the data owner (DO) loses his own secret key, he can't even access his own data, and PKG can still have access to the users data.

In reality we need data owner (DO) should have power to control their own data and distribute secret key to users. To better protect the privacy and guarantee availability of data, we should change data storage from the

centralized cloud storage systems to the de-centralized storage systems, which have certain advantages over traditional cloud storage such as lower prices, large data throughput, and also don't needs to worry about single point of failure.

Wang and Zhang [6] proposed a framework that achieved a fine-grained access control over data in decentralized storage systems, and search the data based on the keywords. They combines decentralized storage system, attribute-based encryption and the Ethereumblockchain to achieve fine grained access control over data. The data is controlled only by his own owner, trusted PKG is no more needed in this system, and the data owner has responsibility of distributing secret key to data users.

The problem with this system is that it fail to implement attribute revocation functionality. Access policies are kept in blockhain. Once it is deployed it cannot be changed or updated. This caused a problem of users attribute revocation. Revocation mechanism is obligatory for any public key encryption schemes which include multiple users, since some secret keys might compromise or the integration of the owner has changed at some point. In real approach, attribute revocation is not only a tough issue in the research but also obligatory to solve for the ABE scheme. Xu and Martin [6] developed a dynamic user revocation and key refreshing (DURKR) model for attribute based encryption in cloud computing. It enables key refreshing, access rights management and revocation mechanism.

2. LITERATURE SURVEY

A. Dynamic User Revocation and Key Refreshing Model

Xu and Martin developed a dynamic user revocation and key refreshing [6] model for attribute based encryption in cloud computing. It can be described as follows.

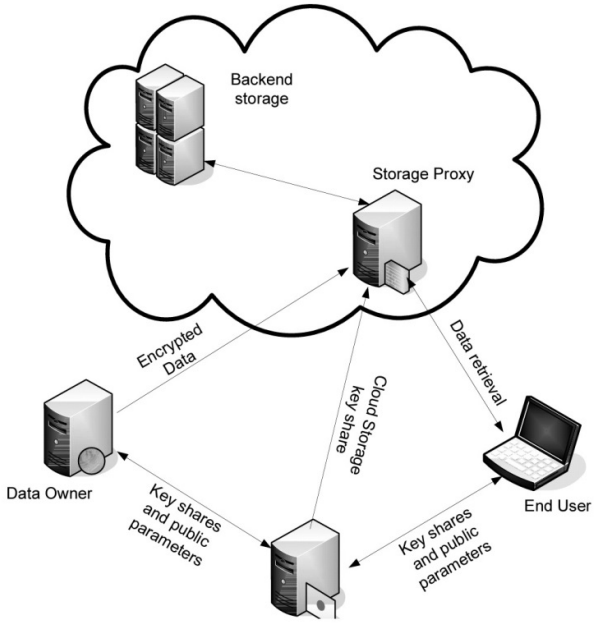

Fig. 1. DURKR model

• **Cloud Services Provider (CSP):** CSP is assumed to be semi-trusted entity responsible for data storage and retrieval service. CSP has a proxy server used for re-encryption of data owners' ciphertexts prior it is sent to data users.

• **Attribute Authority (AA):** AA is the principal trusted element which is responsible for issuing system public parameters, generating attribute key shares and maintaining the master secrets.

• **Data Owner (DO):** DO is responsible for protecting his data by specifying access policies, maintaining user revocation lists, and encrypting data before it is stored to the CSP.

• **Data User (DU):** DU is CSP subscriber whose attributes need to satisfy access policy before the data being decrypted.

To revoke a user from a group, re-encryption is performed on top of CP-ABE scheme to achieve fine grained user level access control. They make use of an delegation attribute, which is assigned to CSP and it is generated by AA. The delegation key is sent to CSP which is used for re-encryption of ciphertext. CSP has only the delegation key and thus it cannot decrypt the encrypted data. Alongside, the delegation key share is used to achieve revocation or system key refreshing. The master secret is divided into two sections. One section is used to generate attribute key shares by the CP-ABE scheme. The second section is used by the CSP (i.e. proxy storage) to issue additional secret share to the users every time when the data is being retrieved. So only the non-revoked users can successfully generate the decryption key. AA re-generates the delegation key share for CSP only when the system key needs to be refreshed. All the key shares and system key are tracked by version numbers, *Vno*, it is initially set to 1. When an event of attribute revocation occurs, it increases by 1.

B. Blockchain based framework for data sharing

Shangping Wang, Yinlong Zhang, Yaling Zhang proposed a blockchain based framework [7] for data sharing with fine grained access control in decentralized storage system framework.They combines decentralized storage system IPFS [8], attribute-based encryption(ABE) technology and the Ethereumblockchain to achieve fine grained access control over data. The data is controlled only by his own owner, trusted PKG is no more needed in this system, and the data owner is responsible for distributing secret key to data users.

The Ethereumblockchain is used to handle the users secret key. Thus the key management problem in the traditional ABE schemes is solved. Whenever a user lost his own secret key, he only needs to decrypts the corresponding transactions data from the Ethereumblockchain, read it and get its own secret key information.

The encrypted keyword indexs are built for the shared file and stored its information on the Ethereumblockchain. Furthermore, the smart contract [9] is deployed on the Ethereumblockchain for implementing the keyword search on the decentralized storage systems. Once the smart contract is deployed on the Ethereumblockchain, it will operate in accordance with the logic and in good faith of the smart contract. The service fee will be paid only if the users gets the correct search results,. The problem in traditional cloud storage that search service providers may not honestly return search results is solved in this scheme.

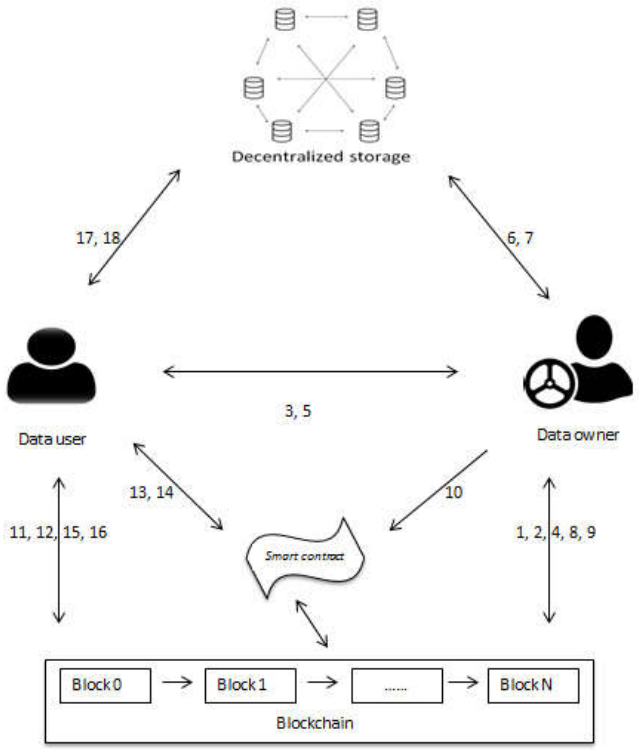

Fig. 2.A Blockchain based framework for data sharing (Architecture)

The description of each step number in Fig. 2.is as follows:

1. *DO* setups the system using master key, encrypt it and then it is embedded into an Ethereum transaction.
2. *DO* deploy the smart contract on the Ethereumblockchain.
3. *DU* sends a registration request to *DO*.
4. *DO* generate secret key for *DU*, encrypt it using the sharedkey and then embed into an Ethereum transaction.
5. *DO* sends the transaction id of secret key, address of smart contract, smart contract source code and smart contract ABI to *DU* through a secure channel.
6. *DO* selects a keyword from the shared file, and uses AES algorithm to encrypt it and upload it to IPFS.
7. *DO* note the file location returned by IPFS.
8. *DO* uses AES key *K* to encrypt the file location returned by IPFS, and uses the ABE algorithm to encrypt AES key *K*. Then *DO* select a AES key *K*1 to encrypt these information and embeds it into an Ethereum transaction.
9. *DO* note the id of the Ethereum transaction and AES key *K*1.
10. *DO* generates encrypted keyword index and stores it to the smart contract.
11. *DU* reads the transaction data of secret key from the Ethereumblockchain.
12. *DU* uses the shared key to decrypt transaction data, and gets the secret key.
13. *DU* generates search token and call the smartcontract.
14. The smart contract searches according to the searchtoken and returns the relevant results.
15. *DU* reads relevant transaction data of search results returned by smart contracts;
16. *DU* decrypts the transaction data.
17. *DU* downloads the encrypted file from IPFS.
18. *DU* decrypts the encrypted file.

3. A PROPOSED BLOCKCHAIN BASED FRAMEWORK FOR DATA SHARING.

In the working of existing system, the *DU* generates search token and invokes the smart contract. The smart contract searches according to the passed token and returns the desired results. *DO*reads relevant transaction data based on searchresults returned by smart contracts. *DU* then decrypts the transaction data and downloads the encrypted file from IPFS. *DU* then decrypts the encrypted file and gets desired file.

Now the data user knows the relevant transaction data based on searchresults returned by smart contracts. This transaction data helps to get all the file related to search token. So In future whenever the data user need the data of same search token, he does not need to perform the search again as this search result are stored in transaction.

The *DU* does not need to satisfy access policy for accessing only transaction data. This has created the problem of attribute revocation. A attribute revoked user can also access the transaction data as it is publicly available. The transaction data gave the actual location of file after decrypting it. The *DU*get the file from IPFS and decrypt it as he is already having the secret key

In the proposed system, we make data user to follow access policy each time before getting the file from storage, Whenever the search is performed the desired files are placed in proxy storage with time stamping and location of proxy storage is placed in transaction data. So the data user will get the file from proxy storage within limited period of time. Whenever the data user need the files of same search token then the previous transaction data related to same search will not be sufficient to give the desired file as it stores the proxy location of file. This makes data user to perform search again i.e. the search will be perform according to the attributes of data users. A attribute revoked user cannot access the file which he can before attribute get revoked. He can only access the file which his current attribute can satisfies the access policy. Thus the problem of user attribute revocation is solved.

The Proposed Architectureof Blockchain based framework for data sharing is as follow:

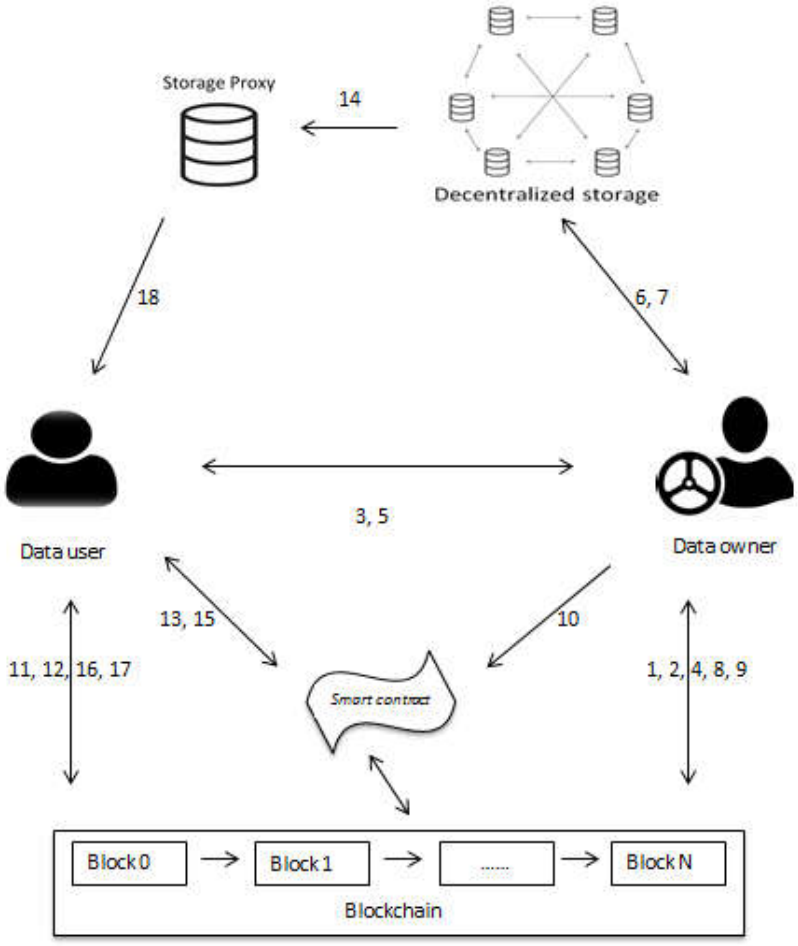

Fig.3. ABlockchain based framework for data sharing (Proposed Architecture)

The working of proposed system is as follow.

1. *DO* setups the system. The system master key is encrypted and then embedded it into an Ethereum transaction.
2. *DO* deploys a smart contract on the Ethereumblockchain.
3. *DU* sends a registration request to *DO*.
4. *DO* generates secret key for *DU*, and uses the sharedkey to encrypt the secret key and embed the encrypted secretkey into an Ethereum transaction.
5. *DO* sends the transaction id related to secret key, smart contract address, smart contract ABI and smart contract source code to *DU* through a secure channel.
6. *DO* selects a keyword set from the shared file, and uses AES algorithm to encrypt the file and uploads it to IPFS.
7. *DO* records the file location returned by IPFS.
8. *DO* uses a selected AES key *K* to encrypt the file location, and uses a selected ABE algorithm to encrypt AESkey*K*. Then AES key *K*1 is selected to encrypt these information and embeds it into an Ethereum transaction.
9. *DO* records the id of the Ethereum transaction and AESkey*K*1.

10. *DO* generates encrypted keyword indexs and stores it to the smart contract.
11. *DU* reads transaction data related to secret key from the Ethereumblockchain.
12. *DU* uses the shared key to decrypt transaction data, then gets secret key.
13. *DU* generates search token and invokes the smartcontract.
14. The smart contract searches according to the search token.
15. The search results are placed in proxy storage.
16. *DU* reads relevant transaction data based on searchresults returned by smart contracts
17. *DU* decrypts the transaction data.
18. *DU* downloads encrypted file from proxy storage and decrypts the encrypted file.

4. CONCLUSION AND FUTURE SCOPE

The ABE technology and searchable encryption technology on ciphertext are important technologies for solving data privacy and fine grained access control problems. The decentralized storage approach can solve theproblem of single point of failure in traditional cloud storage systems. At the same time, compared to centralized storage, it also has a series of advantages such as low price and high throughput. In this framework, no trusted PKG is needed. The data owner has the power to issue secret key for users and encrypts his data under specified access policy to achieve fine-grained access control over data. The keyword search function on the ciphertext is implemented based on the smart contract on the Ethereumblockchain, and the problem that the cloud serverreturn wrong results does not return results or in the traditional cloud storage is solved. In the proposed system, we compulsory data user to satisfy access policy each time before getting the file from storage. The Data user can only access those file whose attribute matches to Data user's current attribute. Thus the problem of attribute revocation is solved. However, our scheme does not implement the functions of access policy update. This is our next research direction.

REFERENCES

[1] **S**. Nakamoto. (2008). *Bitcoin: A Peer-to-Peer Electronic Cash System*. [Online]. Available: https://bitco.in/pdf/bitcoin.pdf

[2] C. Gray. (2014). *Storj Vs. Dropbox: Why Decentralized Storage is theFuture*. [Online]. Available: https://bitcoinmagazine.com/articles/storj-vs-dropbox-decentralized-storage-future-1408177107

[3] A.Sahai and B.Waters, ``Fuzzy identity-based encryption," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer,*2005, pp. 457_473.*

[4] Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute based encryption for fine-grained access control of encrypted data," in Proc. ACM Conf. Comput. Commun. Security, 2006, *pp. 89–98*.

[5] J. Zhang, X. A. Wang, and J. Ma, ``Data owner based attribute based encryption," in *Proc. Int. Conf. Intell. Netw. Collaborative Syst. (INCOS)*, Sep. 2015, *pp. 144_148.*

[6] Z. Q. Xu and K. M. Martin, *Dynamic User Revocationand Key Refreshing for Attribute-Based Encryption in CloudStorage*, In 11th International Conference on Trust, Security and Privacy in Computing and Communications, 2012, *pp.844-849.*

[7] Shangping Wang, Yinlong Zhang, Yaling Zhang," A Blockchain-Based Framework for Data Sharing With Fine-Grained Access Control in Decentralized Storage Systems",Volume 6,in *IEEE Computer Society*, july 30 2018.

[8] J. Benet. (2014). ``IPFS-content addressed, versioned, P2P file system." [Online]. Available: https://arxiv.org/abs/1407.3561

[9] K. Christidis and M. Devetsikiotis, "Blockchains and Smart Contracts for the Internet of Things," *IEEE Access*, report Deloitte Report, *vol. 4*, 2016, *pp. 2292–2303.*