Role Based Smart Contract For Data sharing

Kweka Bruno Joachim*, Kyung-Hyune Rhee**
*Interdisciplinary Program of Information Security, Graduate School,
Pukyong National University.
*Dept. of IT Convergence and Applications Engineering
Pukyong National University

Abstract

The Internet has allowed many things to move fast, including sharing of data, files and others within a second. Many domains use applications range from IoT, smart cities, healthcare, and organizations to share the data when necessary. However, there are some challenges faced by existing systems that works on centralized nature. Such challenges are data breach, trustiness issue, unauthorized access and data fraud. Therefore in this work, we focus on using a smart contract which is used by blockchain platform and works on decentralized form. Furthermore, in this work our contract provides an access to the file uploaded onto the decentralized storage such as IPFS. By leveraging smart contract-role based which consist of a contract owner who can manage the users when access the certain resources such as a file and as well as use of decentralized storage to avoid single point of failure and censorship over secure communication channel. We checked the gas cost of the smart contract since most of contracts tends to be a high cost.

1. INTRODUCTION

The internet has accelerated and stimulates many things to grow. As we have many things connected to each other results to sharing of information. For example, in healthcare where by the hospital records systems can share the patient's data to another stake holder such as insurance company.

In many cases, the concept of data sharing involves three things; one is data requester, data provider and data itself. The advantage of sharing data is to promote collaboration, analyze and utilize data to generate significant value, and maintain accountability and transparency. However misuse of shared data could be resulted into remarkable challenges.

Furthermore the growth of internet has allowed moving from traditional local storage into cloud storage. The cloud storage offers scalability, easy data accessibility and at the low cost. Example of cloud storage companies are google drives, dropbox and many others. However cloud storage possesses challenges such as security concerns, unauthorized access which resulted into data breach and fraud [1]. To address above challenges especially in access control, we propose using smart contract role based built on the blockchain that can achieve and manage users who has the criteria to access as well as to audit any changes occurred.

Blockchain gained popularity because of the bitcoin which was introduced in 2008. The concept of blockchain consists of distributed ledger technology for transactions to be stored globally as blocks [2]. Blockchain technology has changed a lot and recently blockchain has been extended into complex system, which can also be used to transfer a data. For example ethereum use blockchain to transfer data, and also they introduced the concept of smart contract or crypto-contracts, these contracts are logically programmed to execute when certain conditions are met and they can transfer cryptocurrency or data over the blockchain [3]. Since all contracts are well distributed on the public peers and all transactions are known to all peers hence no need of having a third party. Which brings the feature of maintain a permanent audit trail. The immutability features of blockchain provide transparency and protection of an important data.

In this work, we propose a smart contract role based access control based on file sharing case between provider and owner. Where the data owner has the ability to add and remove users whom he wants to.

Our contributions can be summarized as follows:

- We provide a smart contract that has the ability to provide an access control list, gives the owner the power to manage the list of users
- The proposed smart contract approach is flexible to be extended into any contracts that they need to have an access control.

The remains of the paper are constructed as follows: section II is related works that motivate this work. Section III is illustration and detailed overview of the proposed system. Section IV, provides a glimpse of pseudocode of smart contract and gas cost. In section V, we summarize the conclusion of the paper.

2. RELATED WORK

To our knowledge, most literature in smart contract- access control has been articulated in health care and IoT and in organizational level.

RBAC-SC a role based access control the author used a smart contract to achieve the trans organizational utilization of roles. The key idea is to publish all relevant information of user-role assignments in a smart contract and deployed on a blockchain[4].

In the field of IoT they proposed a smart control access control framework, composed of three contracts, one is multiple access control contracts, judge contract and last is register contract with the reason to achieve trust worthy access control for IoT systems and remove a single point of failure since it is occurred much in centralized systems that control access rights to various of objects in IoT which can be compromised with an adversary and caused damages [5].

Another review in healthcare, they named their access control framework Ancile, utilize smart control for tighten access control and data modifications because of exist infrastructure and centralized system in healthcare cause to leaked the patient information and violate data privacy [6].

Although each of the related works explained their intension in relating of using smart contracts but some works consider gas cost others didn't but in this work we focus on expanding smart contract functionality in terms of access control and to be used in case of uploading files and sharing using the decentralized storage like IPFS.

3. OVERVIEW OF PROPOSED WORK

The main focus is to provide the systematic, authentic access using a smart contract, secure and reliable way to enable file sharing between participants. The first step, the data provider or the owner create a smart contract, directly published on the blockchain and receive the contract address. Within a contract the owner can interact with the methods or functions such as add, delete user as way to manage the access control to his data.

File owner

The owner of the file, he can upload the file to IPFS but only give certain user an access to the file. He puts his file in his working directory and encrypts it with user public key and upload to the IPFS storage and ipfs generates the hash of encrypted file. And the file will be available into the ipfs network.

IPFS

We proposed to use IPFS storage because it is expensive to store data on ethereum blockchain. And also the scalability will be a problem. The advantages of using IPFS is huge amounts of data can be stored and just a hash of the file can be stored on ethereum. As for the developer perspective, IPFS provides an APIs which are easy to use to access the ethereum network and IPFs nodes.

USER1

In order for this user data receiver to be able to have an access first, his public address corresponding to the ethereum address must be existing in the file owner contract if not, access denied. That is the first condition, and second he can retrieve the file and decrypt it, by using his private key of the public key that was used to encrypt the file.

Any malicious party cannot decrypt the file because of lack receiver's private key. So far we recommended encrypting the files before upload it and many tools are available to assist on encrypting the files such as GPG [8] which is free tool.

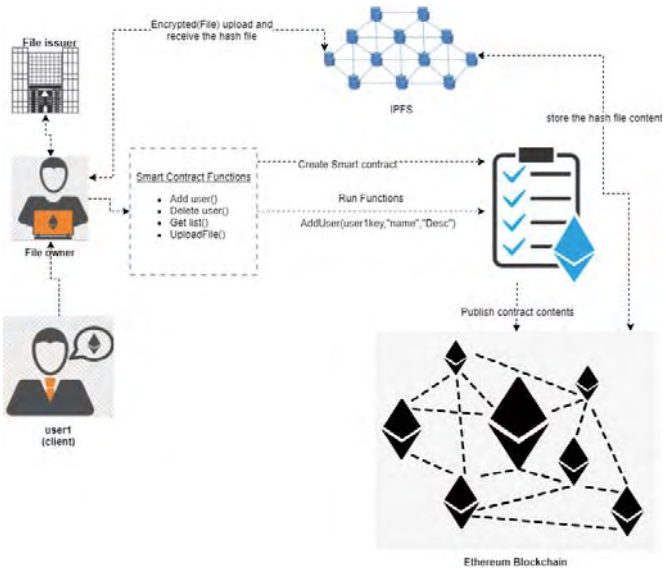

Figure1. Overview of File Sharing

The code below is the code snippet help in making the access control manageable, because only the owner of the contract can add or remove the users. Hence the file owner (owner of the contract) has the ownership of control and manages the data.

```
contract ACLContract {

      address public owner;
      address [] public users;

     function ACLContract() { //constructor
         owner = msg.sender;
         users.push(owner);
     }

     function addUser(address user){
         if(msg.sender != owner) throw;
         users.push(user);
     }

     function getIthUser(uint i) constant returns(address){
         return users[i];
     }
     function getUserCount() constant returns(uint) {
         return users.length;
     }
     function deleteIthUser(uint i) {
         if(msg.sender !=owner) throw;
         delete users[i];
     }

     function isUser (address candidate, string method) returns (bool) {
         for(uint8 i=0; i<users.length; i++){
             if(users[i]== candidate){
                 LogAccess(candidate, now, method,"SuccessFull access");
                 return true;
             }
         }
         LogAccess(candidate, now, method, "Failed Access");
         return false;
     }
     event LogAccess (address indexed by, uint indexed accessTime, string method,
string desc);

```

Figure2. Access control Smart Contract

Our analysis and evaluations we did check our smart contract in terms of gas cost and here are our results;

Table1. Gas Cost of Different Functions in the Smart Contracts

Function	Gas used	USD
Create contract	853318	0.55808
Add user	48782	0.03189
Delete user	13780	0.009

Cost

Ethereum instructions run on gas. Executing a smart contract it cost gas, hence running a smart contract each every instruction costs a certain amount of gas. Designing of smart contract should be very considered in terms of gas cost. In our contract, we use bytes as way to reduce the cost.

As it can been seen in the table1, the highest cost corresponds to the creation and deployment of the smart contract on the blockchain at 0.55808USD. The cost is one-time cost to setup and initialize the contract. The remains of other functions have relatively low cost. Apart from that, these functions the costs will vary as their inputs lengths differ.

Nevertheless the costs can be lowered depends on the system needs and types of data will be used.

Management

In our contract, we use functions as add, delete (revocation of users, in case of user time expired) participants to simplify the tasks of the owner of contract.

Restriction Policy

We restrict only the owner of the contract can be able to perform the contract functions if any of non-owner of the contract will try to perform an execution the process will not be executed.

Confidentiality

Uploading and sharing of data like a file, we use tool like GnUPG to achieve privacy and encryption when upload the files onto the decentralized storage.

4. CONCLUSION

In this paper we discussed and proposed the smart contract role-based (owner of the contract) and we integrate the functionality of contract into the file sharing scenario, where the participants they can leverage the contract methods such as to create an access control with an ability to check the users exist address and allow them to have an access to the files. Also we provide the confidentiality of files by using GnUPG a cryptographic tool.

ACKNOWLEDGMENT

This research was supported by the MSIT(Ministry of Science, ICT),Korea, under the ITRC(Information Technology Research Center) support program (IITP-2018-2015-0-00403)supervised by the IITP(Institute for Information & communications Technology Promotion and partially supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIT) (No. NRF-2018R1D1A1B07048944)

REFFERENCES

[1] Sudha, M., and C. Balakrishnan. "An Analysis on Cloud Data Storage." *International Journal of Computer Science Engineering & Technology* 2, no. 4 (2012).

[2] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." (2008).

[3] Wood, Gavin. "Ethereum: A secure decentralised generalised transaction ledger." *Ethereum project yellow paper* 151 (2014): 1-32.

[4] Cruz, Jason Paul, Yuichi Kaji, and Naoto Yanai. "RBAC-SC: Role-Based Access Control Using Smart Contract." *IEEE Access* 6 (2018): 12240-12251.

[5] Zhang, Yuanyu, Shoji Kasahara, Yulong Shen, Xiaohong Jiang, and Jianxiong Wan. "Smart Contract-Based Access Control for the Internet of Things." *arXiv preprint arXiv:1802.04410* (2018).

[6] Dagher, Gaby G., Jordan Mohler, Matea Milojkovic, and Praneeth Babu Marella. "Ancile: Privacy-preserving Framework for Access Control and Interoperability of Electronic Health Records Using Blockchain Technology." *Sustainable Cities and Society* 39 (2018): 283-97. doi:10.1016/j.scs.2018.02.014.

[7] Benet, Juan. "IPFS-content addressed, versioned, P2P file system." *arXiv preprint arXiv:1407.3561*(2014).

[8] https://gnupg.org/