2020 3rd International Conference on Hot Information-Centric Networking

Blockchain-Based Secure Collaboration Platform for Sharing and Accessing Scientific Research Data

Afnan M. Alniamy
*D*ept. of Electrical Engineering and Computer Science
The Catholic University of America
Washington DC, The United States of America
76alniamy@cua.edu

Hang Liu
Dept. of Electrical Engineering and Computer Science
The Catholic University of America
Washington DC, The United States of America
liuh@cua.edu

2020 3rd International Conference on Hot Information-Centric Networking (HotICN) | 978-1-7281-9216-1/20/$31.00 ©2020 IEEE | DOI: 10.1109/HotICN50779.2020.9350856

***Abstract*—Research teams or institutions in different countries need an effective and secure online platform for collaboration and data sharing. It is essential to build such a collaboration platform with strong data security and privacy. In this paper, we propose a platform for researchers to collaborate and share their data by leveraging attribute-based access control (ABAC) and blockchain technologies. ABAC provides an access control paradigm whereby access rights are granted to users through attribute-based policies, instead of user identities and roles. Hyperledger fabric permission blockchain is used to enable a decentralized secure data sharing environment and preserves user's privacy. The proposed platform allows researchers to fully control their data, manage access to the data at a fine-grained level, keep file updates with proof of authorship, and ensure data integrity and privacy.**

***Index Terms*—Blockchain, Access Control, Attribute-based Encryption, Hyperledger Fabric, Data Management**

I. Introduction

Academic scientific research is becoming more international, which requires experts' involvement, access to specialized equipment, the development of new ideas, and tapping into new sources of funding. Collaborative research between different scientific groups, even between institutions from different countries, is an effective key for successful research implementation. A collaborative platform is an online service that provides a virtual environment to connect multiple researchers for joint research and implementation of scientific projects. There is a strong need for a reliable, private, and user-friendly platform solution that will achieve the goals of researchers' collaboration and data security and privacy. Systems employed for project collaborations are largely centralized with single computing storage that makes it vulnerable to malicious attacks [1]. This collaboration involves the cooperation of multiple researchers at different locations with different roles and obligations. This increases the needs for tracking and sharing the project assets by implementing access control in course level. Applying fain-grained access control over research information or sensitive data is important in this situation. Using Attribute-based Access Control is great for ensuring the fine-grained access control of all stored data. However, this management is still performed through a

centralized service and with the use of Blockchain technology, creates a strong solid feature for a decentralized architecture of collaboration systems and provides unique advantages over centralized data storing issues. Additionally, one of the most important aspects of the Blockchain technology is the degree of transparency that it provides, making users' privacy difficult to preserve. The introduction of permission blockchain networks offers improved privacy by allowing selected participants to join the network, which will help to overcome this problem [2]. While centralization can pose risk to a project, shared collaboration introduces a different risk, since there are many versions of the updated files in the lifecycle of a project, some with unique values that lead to having features that prove the authorship of any important pieces and manage system access permissions. In this paper we propose a Blockchain-based platform for scientific scholars' shared work and collaboration. This platform allows researchers to connect in a collaboration-research community in order to enable easy and secure data sharing in an efficient, fine-grain, transparent and traceable manner. This system design provides a sharing solution for researchers concerned about confidentiality and provides researchers with control over how their data is managed on a fine-grained level. The contributions of this paper are listed below:

- We present a blockchain-based collaboration system that empowers the owners of the research data to prove their ownership, manage secure data sharing, and grant and revoke access to the data
- We define the required features of the system, including the fine-grained access control policy formulated by the owners, the verifiability of all participants' identities, and the decentralized environment to ensure data integrity.
- We propose a ciphertext-policy attribute-based encryption (CP-ABE) mechanism for data access control integrated with the Hyperledger Fabric blockchain to ensure the data integrity and immutability, which guarantees that once the data is stored it cannot be modified afterward.

This paper is organized as follows: Section II reviews the background of blockchain and attribute- based encryption along with related works. In Section III, we explore the use cases and our system requirements. In Section IV, we present the system architecture with the design goals and threat model. In Section V, the design details are discussed. In Section VI,

978-1-7281-9216-1/20/$31.00 ©2020 IEEE

Authorized licensed use limited to: University of Liverpool. Downloaded on June 15,2021 at 03:34:52 UTC from IEEE Xplore. Restrictions apply.

the security of the proposed system is analyzed. Section VII concludes the paper.

II. Background and Requirements

Nowadays, researchers can work with each other for the purposes of scientific research and writing by using modern tools such as social media networks. Sometimes these modern tools allow the researchers to share their manuscripts and drafts in order to perform collaborative work and peer review. The increase of the "Distributed Ledger Technology" and the "Peer-to-Peer content-based systems" caused the modernization in the field of digital security and sharing of content. In connection, this advancement is observed in blockchain technology with the utilization of the concept of the smart contract [3].

A. Blockchain Technology

In 2008, Satoshi Nakamoto [4] proposed a distributed peer-to-peer digital cash system, and this was the cryptocurrency-based implementation, Bitcoin. This technology provides a digital ledger of every record in 'blocks' which are linked together by the cryptographic validation hash value of the previous blocks and ensures that the current block is not tampered with. Blockchain is a distributed ledger in which each peer has a full copy of the ledger. The great reliability of the blockchain is achieved by transparent decentralization, digital signatures, consensus protocol, open-source code, immutability and the tracking of the transactions. All peers in the network must agree on a consensus algorithm to protect the blockchain stored records. The most important consensus algorithm implemented blockchains are: 1. - Proof-of-Work (PoW), proposed by Dwork and Naor [5] which refers to a proof to confirm that a certain amount of work has been done. Bitcoin is based on the PoW consensus algorithm. The system allowed all nodes to calculate a random number fairly when a block is generated. 2.- Proof-of- Stake (PoS), first proposed in PpCoin [18], is mainly used to solve the problem of resource waste in PoW [6]. It is more intuitive to understand from the implementation algorithm formula of PoS, but basically it is based on the idea that the more stakes the nodes have, the greater the chance that a system will succeed.[6]. 3.- Practical Byzantine Fault Tolerant (PBFT) introduced by Lamport, Shostak and Pease [7], is a classic form of Byzantine General Problem (BFT) state machine replication protocol. This algorithm consists of three phases of protocol: pre-prepare, prepare, and commit, which ensure its guarantee. Currently, blockchain is widely used as the underlying data structure and consensus mechanism for financial fields (such as with bitcoin) [4], for alternative cryptocurrencies (such as with altcoins) and for non-financial fields (such as with decentralized application platforms like Ethereum [8] and Hyperledger) [9]. Buletin [7] created Ethereum for deploying decentralized applications on top of the cryptocurrency blockchain network.

B. Hyperledger blockchain platform

Hyperledger Fabric (HF) is a private and permissioned distributed ledger providing an execution environment for smart contracts. It is an open source project founded by the Linux Foundation in 2015. HF is an open source permissioned blockchain designed for enterprises. HF provides components with the definition of roles between the members in the blockchain network and the execution of chaincode that delivers confidentiality, flexibility, and scalability [9]. Hyperledger Composer is an open source project hosted by the Linux Foundation and IBM. It is a set of tools for developing blockchain applications by supporting Fabric blockchain infrastructure and run-time. Each Hyperledger Composer application in the business network is represented by a chaincode process. The chaincode that represents the network contains logic models in JavaScript to execute the transaction. Chaincodes are similar to smart contracts on other blockchain networks. Hyperledger Composer has several components that define data and the transaction logics as the following:

Participants are the members in the network and interact through invoking the transaction.

- Assets are objects stored in the blockchain.
- Transactions represent the actual transaction run by the participant related to the assets.

For more information on Hyperledger Composer applications refer to [10]. The Hyperledger project contestants decided and developed it as information technology. The HLF network nodes are classified into three different types: The requirements are made by the clients for "transactions execution", "processing participation", and "broadcast transactions" to ordering services; Peers are those who perform the processing workflow of the transaction, manage the registry of Blockchain and validate them;

- The "Blockchain Registry" is data structure that records all the transactions like a hash chain and displays the state of ledger.
- There are few numbers of peers that perform the transactions processing decided by their relative chaincode policy. These peers are known as endorsing peers, or sometimes they are simply called 'endorsers.'

All transactions' general orders are established by "Ordering Service Nodes" (OSN). They are also known as orderers in the "Distributed Consensus Algorithm" based Blockchain, in which there is an update of each transaction for the system state, the blockchain history, and the endorsing peers' cryptographic signature. The transaction ordering and peers' separation also makes the modular consensus in HLF a simplification of consensus protocols replacement.

There are many concepts used in order to briefly define the process of business within the HLF and Composer platform framework, including events, transactions, participants and assets. Assets are intellectual or tangible resources, property, or services; registries keep their records. Anything in the business network can be termed an Asset, including a house that is marked for the sale, a sale listing, or a house's land certification of registration; even the insurance documents related to said house can be known as assets with respect

Authorized licensed use limited to: University of Liverpool. Downloaded on June 15,2021 at 03:34:52 UTC from IEEE Xplore. Restrictions apply.

to the business network. There must be a unique identifier of Assets.

C. Attribute Based Encryption

In 2005, Sahai and Water proposed the concept of ABE, which is a one-to-many public key encryption that is a method of access control. It originated from Identity-based encryption [11]. The secret key, the user and the ciphertext are based on a set of attributes such as Name, Position or Location, and if the user's attributes satisfy the access policy set by the encryptor, the decryption is possible. In 2006, [12] Goyal et al. introduced a key policy attribute-based encryption for fine-grained access to encrypted data. KP-ABE forms the attributes on the plaintext and the access policy is embedded with the user's key. The following year, Bethencourt et al. [13] introduced the ciphertext policy attribute-based encryption (CP-ABE), where the ciphertext is associated with access policy and each secret key is associated with the set of attributes. Most of ABE-based schemes and all attributed and distributed keys are managed by an outsource trusted central authority (CA) that is assigned all the private keys for users. And in case of compromising the CA and all stored data, there would be no privacy of the stored data. To address this issue, Lewko and Water [14] proposed multi-authority CP- ABE, where the CA is still available for the initialization phase and upon the user's request, and CA distributes the public parameters, verifies the attribute authorities [15] and [16] proposes multi-authority CP-ABE schemes where the all attribute set is divided into subsets where each subset is maintained by only one authority. These schemes are secure regarding gaining attributes' private keys, however, if any attribute authorities are compromised the chance of gaining access policies is high, and that is the major security problem. The distributed nature of the blockchain allows for the building of an infrastructure to control access and solves the problem of a single point of failure in centralized management. Yuan et al. [17] combined CP-ABE with blockchain storage. Their scheme enables data sharing dependent on attributes but relies on a central authority for the attribute assignment. In addition, their scheme didn't implement the solution of the access policy update. [18] Jamel et al. suggested using the blockchain as an infrastructure for access control management in a sharing system. This system is implemented using a CP-ABE scheme [19] and Multichain platform. Their experimental results indicated that using both technologies provides security and privacy benefits such as auditing, non-reputation, and no single point of failure. However, the scheme depends on a node that is considered to be a synchronization manager and it is responsible for generating the keys, and the nodes in the system verify the legitimacy of the users and add a time dimension to the shared file which is encrypted by CP-ABE. Also, the design requires the data owner to generate two keys for each file, K1 is bind with data user's public key and K2 is encrypted with the timely CP-ABE. Wang et al. [20] proposed a system for data sharing and access control through integrating IPFS decentralized storage, Ethereum blockchain, and attribute-based encryption. In this scheme, the data owner has the right to generate and access the secret keys and the mapping of the private keys is managed by smart contracts in the Ethereum blockchain, however, the scheme did not implement a key revocation mechanism.

D. USE CASE AND SYSTEM REQUIREMENTS

To explain our system requirements, we came up with a few use case scenarios that will be illustrated in the system process under each case. Consider that all researchers and contributors, who are either individuals or organizations, use our collaborative system to share and manage their data and projects. We consider a researcher Alice who wishes to use our system and shares her data; these data might be a project, a paper, codes, or metadata . . . etc., for collaboration, and she is not yet a member of the system

1) Registration Scenario: in order for researcher Alice to start using the system and its services by sharing, storing and managing her data, and to be accessible to other contributors, she needs to register with the system.
2) Data Access Scenario: giving Alice all the rights to manage her data in the system after uploading and sharing the project so she can access, check, track her project at any time.
3) Grant/ Revoke Scenario: after uploading the project, Alice can allow any contributors to be a part of her project and give them the role of read, write, and/or delete permission; and she can withdraw this permission as well.
4) Renew/Revoke Key Scenario: After a while Alice needs to change the encryption key or access policy of a file for the sake of safety.
5) Update Project Scenario: After uploading the project Alice reviewed the data and found errors that need to be fixed.
6) Remove Project Scenario: Alice accidentally uploaded a duplicate copy of one project, and she wishes to delete it.

E. System Requirements

Based on the use cases, we have created a set of requirements that our system must satisfy:

Access Rights: The right given by the data owner, "Researcher," to other users, "Contributors," to view the requested project is controlled by the owner. Our system should empower the researchers by enabling them to have full and complete control over their data project "Assets" and have a more active role in managing them. In addition, the system should provide the ability to grant/revoke access to the project based on which contributors can access the data. The administering of the shared projects is handled by the researchers who are the owners of that data, so that they no longer need to request access from authorities.

Security and privacy: our system should preserve the confidentiality of the shared data and maintain the privacy of the system members as well. This includes the enforcement of several access control rules. In addition, in case of an attacker

Authorized licensed use limited to: University of Liverpool. Downloaded on June 15,2021 at 03:34:52 UTC from IEEE Xplore. Restrictions apply.

who is trying to access or obtain any shared data from the system, the data must remain secure and protected.

Performance: our system should provide the capability of operating all the functions such as accessing the project and managing authorization at any time.

Efficiency: our system should be scalable enough to deal with any number of participants and achieve availability by allowing them to access their data and avoid a single point of failure.

TABLE I: Notations table

Notation	Description
AAS	Authentication Authority Server
SK	Secret Key
Kc	Encryption key
Ka	Authentication Key
PK	System Public Key
MSK	System Master Key
S	Set of Attribute
T	Tree access structure
PubK	Participant public key
PrivK	Participant private key
IDr	Researcher identity
IDc	Contributor identity
K	symmetric key
EncKc	AES encryption algorithm
DecKc	AES decryption algorithm
Kc'	Encrypted symmetric key K by CP-ABE

III. System Architecture

In this section, we present the system and threat models, then an architecture of it including the way that it is achieved the requirements described in the previous section.

A. Design Goals

This system is a Blockchain network composed of multiple entities which are:

Researchers: this entity is considered to be the owner of the data who wishes to share his projects in the network. The Researcher can upload their data to be stored and query the blockchain for information regarding the tracking and auditing. However, researchers are required to encrypt on their client side before uploading and decrypting on their end, after retrieving it from the blockchain.

Contributors: any members in the network who wish to access the data or contribute as a part of a project; a researcher could be a contributor if her wish is to work in others' project. Contributors are able to retrieve data after they are allowed, and they need to decrypt the data on their client side. They can perform some operations on the data if they are granted write, update, and/or delete rules.

Network Administrator: this entity is responsible for maintaining the blockchain peer nodes on the network and managing participants' [researchers and contributors] identities.

Hyperledger Fabric Blockchain Network: it is the core component of our system; it connects peer nodes with shared ledger. Participating nodes are responsible for storing data, executing smart contracts, validating transactions and committing new blocks to the network.

Certificate Authority (CA): it is responsible for issuing a digital identity for participants on the blockchain network as members or peer nodes. The digital identity is attached to transactions invoked on the blockchain and it allows verification of the sender's identity and the integrity of the block.

Authentication Authority Server: this server is responsible for storing cryptographic data that will be used to authenticate the participants and connect them to the blockchain.

Client-Side App: the client service is the point that connects the participants with the blockchain network. It submits transactions to execute operations (read, create, delete, update) to the ledger. Any network's participants who wish to access the client-app must provide login information. In addition, different participants in the network will have a different client App that operates and performs functions based on their roles. For example, network admin is the only member who can register new researchers and contributors.

B. Threat Model

The entities in the system may threat the system in the following ways:

The Authentication Authority Server: The AAS, which is in charge of storing cryptographic materials as secret keys may cause the leakage of keys and learning the secured data.

Researchers: data owners might want to repudiate the data record in the blockchains, or they may tamper the files stored in the storage servers.

Contributors: as the requestors of the stored files may try to decrypt those files which they have no legal access to; they are with level of authorized access to the blockchain and try to perform operations they are not entitled to.

External Attackers: any attackers who do not have access authorization to stored data or do not have an identity in the network. Attackers may also try to compromise network nodes to gain access to data in the blockchain.

C. Proposed System

To address the requirements that are described in this section, we proposed access control management for research sharing and collaboration platform building on the top of blockchain network. Figure. 1 illustrates architecture overview of the collaborative system and the interactions between researchers and contributors for stored files in the Blockchain. Our system provides availability requirements by adopting blockchain technology that allows the system to be fully available at any time and ensure that all nodes are in the network to prevent a single point of failure attack. We use a Hyperledger Fabric private blockchain that requires any participants in the network to be authenticated and authorized for intended activities. It requires that each peer in the network with responsibilities is identifiable, which reduces any kind of node misbehaving or unavailability. This private blockchain is known for performing transactions faster than the public network. The most appealing aspect of blockchain is the

Authorized licensed use limited to: University of Liverpool. Downloaded on June 15,2021 at 03:34:52 UTC from IEEE Xplore. Restrictions apply.

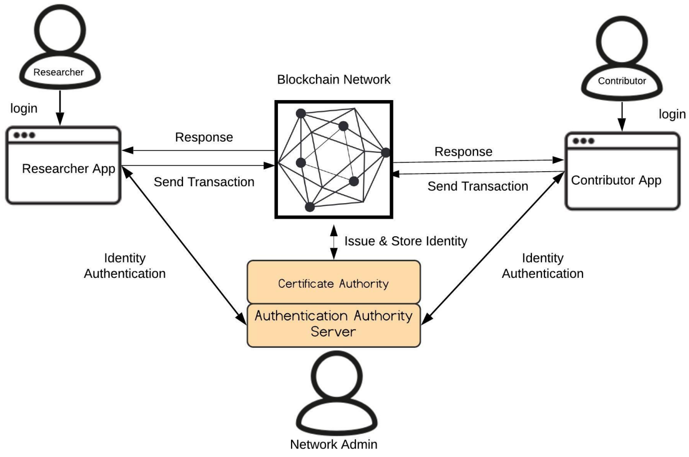

Fig. 1: illustrates our system architecture. Researcher encrypts and stores the data on the blockchain. Contributor can interact with blockchain by invoking transactions or query the ledger through his Client app

degree of transparency it provides; we use Certificate authority to issue participants' identities and key pairs, which will be used to sign each transaction before publishing to the blockchain ledger, in order to keep data confidential and protect participants' privacy. One of the blockchain's features that is used in our system is the Smart contract's, "Chaincode" in Hyperledger Fabric blockchain used to enforce the access control rules. The use of chaincode with its cryptographic techniques will provide the capability to facilitate data sharing and manage the researchers' granting/revoking access to contributors.

Considering the Registration scenario, where Alice would like to join the network and complete the registration process; she will get a digital identity and key-pair from the Certificate Authority CA and it will be stored in the Attribute Authentication Server AAS securely. After being a member in the network as a researcher, she will share her data project in the network. In order to give a contributor, Bob, access to one of the projects, he needs to request grant access for the intended data. Bob will follow the same registration process to be a member in the system, then submit an access request transaction to Alice. If Bob is granted access to the project, he will query the blockchain for the data he needs, download, and decrypt the data.

IV. SYSTEM DESIGN

In our scheme, we assume that the files uploaded by researchers are authentic and secure. However, researchers will perform the encryption process using CP-ABE before uploading the cipher to the network. In our system we adopted the definition and construction of the CP-ABE scheme [13] to illustrate the construction of CP-ABE, as most of the state-of-the-art literatures adopting CP-ABE for data access control are based on this construction. In this section, we explain the system process that will enable sharing and collaboration amongst researchers and contributors. Our technique is not limited in any way to the research sharing application and it can be applied to any other sharing application. Table. I presents some of the notations used throughout the paper

TABLE II: Structure of Record-Blockchain

Record Id	Researcher PK	File Id	Kc	Hash Value	Contributor Id
01	R005	file1	000000	hgtf54...	C001

A. *Registration and Enrollment*

The starting point of the registration process that is depicted in Figure.2 is a client application, where the researcher's first request is to be registered in the network and get the digital certificate. For the purpose of anonymity and verification we adopted a Key Derivation Function (KDF) also called the key stretching algorithm. KDF is a major part in peer-to-peer and blockchain applications, that takes the password, a salt value, and some cost parameters as input and generates a cryptography strong and hardened authentication key. We are using the password strengthen protocol is to not allow the AAS server to learn anything about the entered password except whether the password is correct or not, where the participant authenticate himself to the server directly during the attempting to login into the system. The registration processed after providing all information needed from the researcher, the applied KDF in the researcher's client service, will use the password and generate all required keys to the user as, key Kc, SK, key-pair (PubK, PrivK). Then these keys are uploaded to the AAS server to complete the registration request. The server will generate the researcher identifier, IDr, using the submitted information as researcher attributes S, will submit IDr, PubK, S to Certificate Authority (CA) and will start the enrollment process. After enrolling the researcher

Authorized licensed use limited to: University of Liverpool. Downloaded on June 15,2021 at 03:34:52 UTC from IEEE Xplore. Restrictions apply.

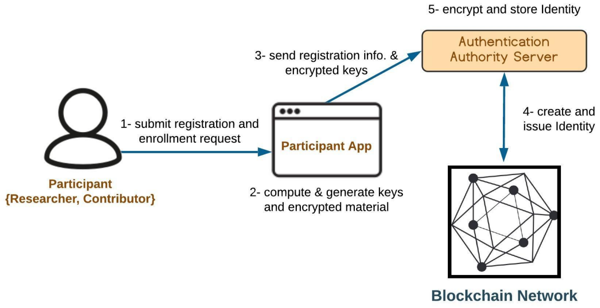
5- encrypt and store Identity
Authentication Authority Server
3- send registration info. & encrypted keys
4- create and issue Identity
1- submit registration and enrollment request
Participant App
Participant {Researcher, Contributor}
2- compute & generate keys and encrypted material
Blockchain Network

Fig. 2: Registration and Enrollment process

identity via CA, the digital certificate and signing key will be issued.

B. Add File

The process of adding files to the network consists of performing three algorithms:

1) Symmetric Encryption: a researcher will encrypt the file before uploading it to the storage using AES symmetric encryption. The researcher will use his Kc as the encryption key.

$$Enc(File) = EncKc(File) \tag{1}$$

2) Ciphertext Policy Attribute-based Encryption: to ensure the fine-grained access control policy, the researcher will utilize the access policy associated with Kc using CP-ABE. Here we adopt the definition and construction of the CP-ABE scheme [13] to illustrate the access structure, where the algorithm encrypts Kc under tree access structure T.
3) Record Kc' to the blockchain: this is done by invoking AddFile transaction, where the smart contract on the blockchain will verify the submitted information as Kc', hash value, timestamp, researcher's PubK and the signature in the form of a record. Table II illustrates an example of the recode uploading in the blockchain ledger with the field related to the data file.

$$AddFileR1 = TS, fileId, Kc', HashV, RpubKSig \tag{2}$$

C. Access Request

This request process shown in Figure.3 performs when a contributor wishes to access a project file and be part of the work. In order to view the file they must have access to the Kc' associated with this file. The contributor will use his client application to submit the AccessFile request to the blockchain peer's node as a transaction. The peer node will execute the transaction and send an event notification, RequestAccessFile, to the researcher who owns the project. If the researcher agrees to grant access to the contributor, he will grant permission through the ShareKey transaction, adding the contributor's PubK to a list, ContributorList, as a reference that proves this user is contributing in the project. The peer node will emit the notification, RequestedShareKey event, and it will display it in the contributor application.

D. Access Revocation

At any time, if the researcher wishes to revoke access to his project from an unneeded contributor, he needs to submit a RevokeAccess to the blockchain indicating the FileId and the contributor identity, PubK. Once the transaction is received, the PubK that added to the ContributorList will be removed and the notification, RevokedAccessRequest, will be displayed in the researcher app.

E. Prove of Ownership

During shared collaboration between contributors there will be many changes and updates to the project lifecycle. In this case contributors would like to prove their authorship regarding the part they worked on. Our system provides a strong mechanism for achieving that. We built a chaincode, CreateFileRecord, that will be invoked by the contributors to record the ownership of their work. From the contributor API, the file hash and the signature will be computed by using the contributor's private key, then it will be submitted to the fabric network. The signature and the timestamp of the transaction data and the hash of the file will be stored in the blockchain.

To verify the file ownership, the application will prove that using the data file and the contributor's certificate to compute and retrieve the hash of the stored file. The validation step will be performed using signature and public key in the that certificate. With this mechanism, any participants in the network can verify that the requested file is been signed by the original editor or contributor and it can be retrieved from the ledger.

V. Security Analysis

We now demonstrate that our scheme meets all the requirements below

Confidentiality: External attacker may compromise one or more nodes for gaining access to the data, our system ensures the confidentiality of all stored files. Using symmetrical encryption to prevent storing the data as plaintext in the storage

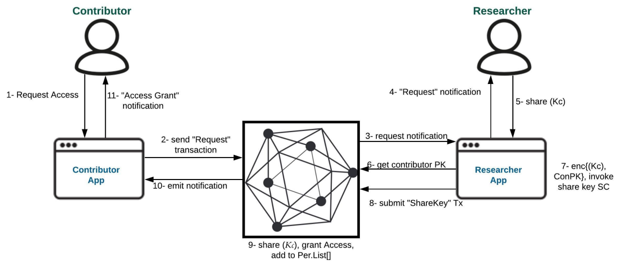
Contributor
Researcher
1- Request Access
11- "Access Grant" notification
4- "Request" notification
5- share (Kc)
2- send "Request" transaction
3- request notification
6- get contributor PK
Contributor App
Researcher App
10- emit notification
8- submit "ShareKey" Tx
7- enc{(Kc), ConPK}, invoke share key SC
9- share (Kc), grant Access, add to Per.List[]

Fig. 3: Request to Access a File

Authorized licensed use limited to: University of Liverpool. Downloaded on June 15,2021 at 03:34:52 UTC from IEEE Xplore. Restrictions apply.

server directly. If attacker tries to access the ciphertext file, he will need to get the Kc and decrypt the Enc(File), and the premise of getting Kc is to be authorized to lookup in the blockchain to find it.

Fine-Grained Access Control: The main function of our system lies in individualized access policy granted by the owners themselves and permission mechanism. Researchers will formulate access policy and apply them to their files, and when the contributor's set of attributes satisfies the policy, they will be able to decrypt the file. In addition to the access policy structures, our system applies permission list smart contract that enforces the access rules and in case of unauthorized contributors attempt to access the Kc, the smart contract will check whether this contributor is in the access list or not.

Ownership Verification: With many changing and updating to the data file, it is important to keep tracking of every contributor work by proving the ownership. Our system creates a smart contract CreateFileRecord that is invoked by the contributor and it computes the file hash and signature using the contributor's private key. In order to verify the ownership, the smart contract will use contributor's certificate, signature, and the timestamp.

VI. CONCLUSIONS

In this paper, we presented a collaboration platform that allows researchers to collaborate and share their data in a secure, transparent, and traceable manner. Our scheme provides fine-grained access control and data management with which the researchers are able to securely share their data online. The cryptographic techniques such as CP-ABE and blockchain smart contract are used to give the researchers full control of the access to their data by granting and revoking permission to any file based on the attribute-based policy while storing and sharing the data with privacy preserving. We analyze the security of our collaboration system and show that it maintains confidentiality, fine-grained access control, and ownership verification.

REFERENCES

[1] G, MENG, Y, LIU, and J, ZHANG, "Collaborative Security: A Survey and Taxonomy", ACM Computing Surveys. PP. 48, July, 2015.

[2] R, Zhang, R, Xue, and L, Liu. "Security and Privacy on Blockchain", ACM Computing Surveys. Vol. 52, PP.1-34, July, 2019.

[3] I.Bentov, A, Gabizon, and A, Mizrahi, "Cryptocurrencies Without Proof of Work". Financial Cryptography and Data Security. Vol. 9504, PP. 4-10, August, 2016.

[4] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, [online] Available: http://www.bitcoin.org/bitcoin.pdf. 2009.

[5] C, Dwork and N. Naor, "Pricing via Processing, Or, Combatting Junk Mail, Advances in Cryptology". CRYPTO'92: Lecture Notes in Computer Science No. 740. Springer: 139–147. 1993

[6] Y. Gilad, R. Hemo, S. Micali, G. Vlachos and N. Zeldovich, "Algorand: Scaling Byzantine Agreements for Cryptocurrencies", Cryptology ePrint Archive Report 2017/454, 2017.

[7] L. Lamport, R. Shostak and M. Pease, "The Byzantine Generals Problem", ACM Trans. Programming Languages and Systems, vol. 4, no. 3, pp. 382-401, July 1982.

[8] V. Buterin, G. Wood and J. Wilcke, "Ethereum White Paper: A Next-Generation Smart Contract and Decentralized Application Platform", [online] Available: https://github.com/ethereum/wiki/wiki/White-Paper.

[9] HyperLedger Fabric, [online] Available: https://www.hyperledger.org/projects/fabric.

[10] Introduction — Hyperledger Composer. [online] Available: https://hyperledger.github.io/composer/latest/introduction/introduction.html.

[11] A. Sahai and B. Waters. Fuzzy Identity Based Encryption. In Advances in Cryptology - Eurocrypt, volume 3494 of LNCS, pages 457-473. Springer, 2005.

[12] V. Goyal, O. Pandey, A. Sahai, and B. Waters. Attribute Based Encryption for Fine-Grained Access Conrol of Encrypted Data. In ACM conference on Computer and Communications Security (A CM CCS), 2006.

[13] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-Policy Attribute-Based Encryption", Proc. IEEE Symp. Security and Privacy (SP '07), pp. 321-334, 2007.

[14] A. Lewko and B. Waters, "Decentralizing Attribute-Based Encryption", EUROCRYPT: Proc. 30th Ann. Int'l Conf. Theory and Applications of Cryptographic Techniques: Advances in Cryptology, pp. 568-588, 2011.

[15] M. Chase. Multi-authority attribute-based encryption. In (To Appear) The Fourth Theory of Cryptography Conference (TCC 2007), 2007.

[16] H. Lin, Z. Cao, X. Liang and J. Shao, "Secure Threshold Multi-Authority Attribute Based Encryption without a Central Authority", Proc. Progress in Cryptology Conf. (INDOCRYPT), pp. 426-436, 2008.

[17] C. Yuan, M. Xu, X. Si and B. Li, "Blockchain with accountable cp-abe: how to effectively protect the electronic documents", 2017 IEEE 23rd international conference on parallel and distributed systems (ICPADS). IEEE, pp. 800-803, 2017.

[18] M. Jemel and A. Serhrouchni, "Decentralized access control mechanism with temporal dimension based on blockchain", Proc. IEEE 14th Int. Conf. E-Bus. Eng. (ICEBE), pp. 177-182, Nov. 2017.

[19] M, Santos, and E, Moura . Hands-On IoT Solutions with Blockchain. 2019.

[20] S. Wang, Y. Zhang and Y. Zhang, "A blockchain-based framework for data sharing with fine-grained access control in decentralized storage systems", IEEE Access, vol. 6, pp. 38437-38450, Jun. 2018.

Authorized licensed use limited to: University of Liverpool. Downloaded on June 15,2021 at 03:34:52 UTC from IEEE Xplore. Restrictions apply.