B-Box - A Decentralized Storage System Using IPFS, Attributed-based Encryption, and Blockchain

Van-Duy Pham*
Hanoi University
of Science and Technology
Hanoi, Vietnam
duy.pv150632@sis.hust.edu.vn

Canh-Tuan Tran*
Hanoi University
of Science and Technology
Hanoi, Vietnam
tuan.tc154144@sis.hust.edu.vn

Thang Nguyen
Hanoi University
of Science and Technology
Hanoi, Vietnam
thang.nguyen@v-chain.vn

Tien-Thao Nguyen
Hanoi University
of Science and Technology
Hanoi, Vietnam
thao.nguyen@v-chain.vn

Ba-Lam Do
Hanoi University
of Science and Technology
Hanoi, Vietnam
lamdb@soict.hust.edu.vn

Thanh-Chung Dao
Hanoi University
of Science and Technology
Hanoi, Vietnam
chungdt@soict.hust.edu.vn

Binh Minh Nguyen†
Hanoi University
of Science and Technology
Hanoi, Vietnam
minhnb@soict.hust.edu.vn

* these authors contributed equally; † corresponding author

***Abstract*—In recent years, centralized storage systems have been extensively adopted by many companies, organizations, and individuals for storing and sharing data. These systems, however, make concerns for users of a single point of failure and the involvement of a centralized entity or third party. Therefore, there is a need for developing decentralized storage systems to overcome the drawbacks of traditional approach. In order to enhance secure and transparent characteristics of decentralized storage systems, in this paper, we present a combination of IPFS (InterPlanetary File System), ABE (Attribute-based Encryption), Multi-Authority ABE (MA-ABE), and Ethereum blockchain. In particular, we facilitate the advantages of IPFS network to store user's data in a distributed manner. Furthermore, we make the use of MA-ABE to encrypt a document, which an user needs to share it among multiple organizations. The hash returned by the IPFS network will be stored in the Ethereum blockchain network to provide trustworthy for all users participating in our system. To the best of our knowledge, it is the first storage system using IPFS, ABE, MA-ABE, and blockchain technologies together to ensure decentralized, secure, and transparent characteristics for storing and sharing data.**

***Index Terms*—InterPlanetary File System, IPFS, attribute-based encryption, ABE, Multi-Authority attribute-based encryption, MA-ABE, blockchain, blockchainize, Ethereum, decentralized storage system**

I. Introduction

In recent years, cloud storage systems have been widely used to meet demands for companies, organizations, and individuals in terms of data storage and sharing [1]–[3]. Users investing in these kinds of services must fully put their trust in those companies to keep their important and private data secured for a long period of time. However, centralised systems tend to pose a threat of being attacked or services being interrupted heavily. Moreover, information leakage cases such as Facebook-Cambridge Analytica [4], [5] have led to a strong movement from using centralised to decentralized data storage systems at the time.

Decentralised storage systems occupy promising advantages over the centralised one, as it has solved successfully the single-point failure problem. In addition, they facilitate more benefits such as low-latency, low price, and completely removing trust in a third party. Recently, Inter-Planetary File system (IPFS) [6], which is a decentralised file system has been receiving wide interest from research and development communities because its foundation is built from a combination of salient research along with ideas of most popular storage systems including Git, BitTorrent, Distributed Hash Tables (DHTs) and Smart File System (SFS). IPFS is a peer-to-peer version-controlled system that provides high-throughput and decentralized distribution.

Although, IPFS establishes a solid foundation for sharing and storing data, applying IPFS into real problems still requires a lot of researches and extensions to meet specific features of applications in practice. First, how to allow users to safely and conveniently share data for multiple users in different organizations? For this problem, in traditional storage systems, cryptographic techniques have been widely used [7], [8]. For example, Attributed-based Encryption (ABE) [8]–[10] and MA-ABE [11] help reduce the number of times encrypting the same data when the data needs to be shared inside an organization or among different organizations, respectively. In both cases, only one version of the ciphertext is stored. Applying ABE and MA-ABE into IPFS, therefore has potential to enhance secure and convenient features of decentralized storage system. Second, how to create undeniable proofs of data sharing in case that users do not trust each other? In centralized storage systems, participants need to believe a third party to store immutable proofs. However, depending on a single party is a drawback that users tend to avoid. Blockchain [12]–[14] is a recent emerging technology that shares a distributed ledger among all participants in the network. This technology is used to address issues related to trust, privacy, and data sharing [15]–[17]. Among blockchain platforms, Ethereum is one of the most widely-used platforms in the world [18], [19]. As a result, combining Ethereum into IPFS network can increase the transparent aspect of decentralized storage.

In this paper, we introduce a data storage system named B-Box that provides decentralized, secure, and transparent characteristics. In particular, we make the use of most promising and effective technologies to build this system. B-Box is available at https://v-chain.vn/solutions/b-box. First, we rely on advantages of IPFS network to store data of users in a decentralized manner. Next, when users need to share a document, we encrypt the data using (MA-)ABE encryption. There are two different cases, covering: (i) if the document is only shared inside an organization, it will be encrypted based on ABE; and (ii) if the data needs to be shared between multiple users in different organizations, B-Box makes the use of MA-ABE. Finally, Ethereum network is used to keep the hash of the data in a unmodified way. To the best of our knowledge, it is the first decentralized storage system that combines IPFS, ABE, MA-ABE, and Ethereum at the time.

The reminder of this paper is organized as follows. Section II provides background information of IPFS, ABE, MA-ABE, and Ethereum. Section III discusses related work. Next, we introduce our system in Section IV. Section V illustrates two use-cases of B-Box in practice. Finally, we conclude with an outlook on future research in Section VI.

II. Background

A. Blockchain technology and Ethereum

Blockchain [12]–[14] is a distributed ledger containing a collection of blocks. In which, each block is composed by transactions and includes a hash of the previous block. This mechanism helps create a chain of blocks, and a block except the genesis block is hooked its parent block. As a result, blockchain technology ensures the immutability of the data because changing data in one block will affect all next blocks. To add new block, blockchain often relies on a set of nodes, which provide the computing power to achieve consensus. A candidate block must be validated by a consensus algorithm before it is added to the chain. Some common consensus algorithms are proof of work, and proof of stake [13], [20], [21].

Ethereum [18], [19] is a decentralized, open source, public platform based on blockchain technology. The structure of the Ethereum is very similar to the other blockchain platforms. The platform contains a shared record of the entire transaction history. Every node on the network stores a copy of this history. It has a feature called smart contract, which facilitates online contract agreements. Smart contract is a small piece of code that operates independently on the blockchain platform without any possibility of censorship, decommissioning, fraud or third-party intervention. The platform includes a fully Turing virtual machine - Ethereum Virtual Machine (EVM) [22], which can execute scripts using an Ethereum computer network. Ethereum also offers a cryptocurrency called ”Ether” which can be transferred between accounts and used to pay miners to help with the calculations. ”Gas” is an internal transaction pricing mechanism, used to minimize spam transactions and allocate resources across the network.

B. IPFS

Interplanetary File System (IPFS) [6] is a decentralized data management system designed based on a peer-to-peer network model. Instead of being location-based, IPFS addresses a file by a content identifier (CID), which is a cryptographic hash of the content of the file. Moreover, IPFS uses distributed hash table (HDT) [6] to support the process of routing and retrieving content from nodes in the network, while using a data structure called Merkle DAG [23] to describe the file as a whole and reconstruct any file from its chunks. IPFS uses InterPlanetary Name Space (IPNS) [6] as a content creation and update system, Interplanetary Linked Data (IPLD) [6] for data management and Bitswap [6] to send and receive notification between nodes. By using a decentralized architecture, IPFS has solved the problem of a single point of failure. In addition, the issues relating to security and data downloading speed have also been significantly improved. Besides, IPFS clusters can configure data replication mechanisms to satisfy the important needs of users. In fact, many applications that use some special technologies such as blockchain cannot store large data such as images or videos due to storage and cost issues. Instead, they use IPFS as a place to directly store the data, and their CID will be stored on the blockchain network. Currently, the combination of IPFS and data encryption is considered as the optimal solution for data access management. Encrypted data will be uploaded to the IPFS platform, while decryption and accessibility to such data are restricted to those who have sufficient access rights.

C. Attributed-based Encryption

Attribute-based Encryption (ABE) [8]–[10] is an access control technology using encryption and decryption. Instead of encrypting for users individually which is slow and inefficient, ciphertext is used to encrypt for multiple users instantaneously. Indeed, private keys of users and ciphertexts are connected to the attributes of desired users or organizations. Decryption to receive plain texts is only possible to those containing sufficient attributes that satisfy a given policy. ABE consists of two types, which are ciphertext policy attribute based-encryption (CP-ABE) and key policy attribute-based encryption (KP-ABE). The main difference between these two types is the method for connecting attributes and policies using private keys and ciphertexts. On one hand, in KP-ABE, an access policy is linked directly with an user’s private key, while the set of attributes of that user is connected with the encrypted data. As a result, with KP-ABE, the owner of the encrypted data cannot control which users that are capable of decrypting such data. Indeed, he can only create his own attributes that link to the ciphertext, and he must trust the party that delivers his keys to permit or prohibit other users when it comes to decryption. On the other hand, in CP-ABE, the access policy is linked to the ciphertext meanwhile connecting attributes of the user with his private key. Therefore, CP-ABE is more favorable in comparison with KP-ABE because of the ability to control and manage those who can decrypt. Accorrding to [24], CP-ABE has four algorithms:

Setup(λ, U) → *PK,MSK*
The setup algorithm takes security parameter and attributes as input. It outputs the public parameters *PK* and a master key *MSK*.

Encrypt(*PK, p, m*) → *CT*
The encryption algorithm takes as input the public parameters *PK*, a message *m*, and an access policy *p* created over attributes. The algorithm will produce a ciphertext *CT*.

KeyGen(*MSK, S*) → *SK*
The key generation algorithm accepts the input including the master key and a set *S* of attributes. It creates an private key *SK*.

Decrypt *(PK, CT, SK)* → *m* The decryption algorithm receives the input as a ciphertext *CT*, access policy *p*, and a private key *SK* linked with attributes. Attributes that satisfy the policy *p* are be able to decrypt the message. The algorithm will decrypt the ciphertext and return a message *m*.

However, the CP-ABE model can be problematic when the authority is attacked or the authority can also become a bottleneck for the entire system due to management and handling of all user attributes. To address these issues as well as enhance data privacy of users, Allison et al. [25] proposed a new mechanism called Multi-Authority Attribute-Based Encryption (MA-ABE). MA-ABE does not exist in a central authority. Each authority manages its own set of properties. Any organization can join to become an authority simply by creating a public key and issuing a private key to users. A user encrypts data through a policy that is built base on attributes. Our proposed system is using CP-ABE and MA-ABE as data encryption mechanisms. Allison et al. [25] showed five algorithms in MA-ABE:

Global Setup(λ) → *GP*
The global setup algorithm takes in the security parameter λ and outputs global parameters *GP* for the system.

Authority Setup(*GP*) → *PK, SK* The authority has to setup the algorithm with *GP* as input to produce its own secret key and public key pair *SK, PK*. *PK* is published public whereas *SK* is kept secret.

Encrypt($m, p, GP, \{PK\}$) → *CT* The encryption algorithm takes in a message *m*, an access policy *p* that is created over attributes, the set of public keys for relevant authorities, and the global parameters. It outputs a ciphertext CT.

KeyGen(*GID, i, SK, GP*) → $K_{i,\ GID}$ The key generation algorithm takes in an identity *GID*, the global parameters, an attribute *i* belonging to some authorities, and the secret key *SK* for this authority. It produces a key $K_{i,\ GID}$ for this attribute.

Decrypt(*CT*, $\{K_{i,GID}\}$, *GP*) → *m* The decryption algorithm receives the input including the global parameters, the ciphertext, a collection of keys corresponding to attributes, and GID. It decrypts successfully when the collection of attributes satisfying the access policy. Otherwise, decryption fails.

III. Related Work

We organize the related work in the area of data storage into two main categories, namely research on IPFS network and data encryption.

Within the former group, a significant number of researchers have focused on combining advantages of IPFS network and blockchain network to reduce the size of stored data as well as to allow users to store multimedia data on the blockchain network. Norvill et al. [26] introduced a system that stores the bytecodes of Ethereum contracts in IPFS instead of on the blockchain directly. By moving codes off-chain and storing the their hash, the size of chain is reduced significantly. Similarity, Zheng et al. [27] proposed an IPFS-based storage mechanism for Bitcoin network. Transaction data will be stored in the IPFS, and only the IPFS hash is packed in the block to reduce the storage size. Xu et al. [28] presented a social network based on Ethereum and IPFS. The system allows users to upload their files such as pictures or videos to an IPFS data storage and receive the corresponding IPFS hash in return. Next, the hash together with user's tweet are stored in Ethereum network. Nizamuddin et al. [16] developed a framework for version control and sharing document using Etherum and IPFS. IPFS is used for storing and sharing a huge number of files with high throughput [6] whereas Ethereum is used to ensure trust for all entities participating the chain.

Research on data encryption for storage systems has also attracted a larger number of researchers. Approaches range from homomorphic encryption, attribute-based encryption (ABE), searchable encryption, to broadcast encryption, etc. [7]. In these approaches, ABE is one of the most interested cryptographic technologies [8]. ABE-based approaches for cloud file systems include KP-ABE [2], [29], [30], CP-ABE [8], and Multi-authority ABE [11], [31].

There is an research [3] published in 2018 that combines IPFS, ABE, and Ethereum. Compared with this contribution, our system not only uses ABE technology but also supports MA-ABE technology. As a result, B-Box can provide a better support to users when they need to safely share a document to users belonging different organizations.

IV. System Architecture

Figure 1 describes the overall architecture of B-Box. In our overall design, we define two main component groups: the Network of Organizations (CA1, CA2, ...) and the Blockchain network, such as Ethereum. An organization can easily join in the Network of Organizations and connect with each other. Each organization can install its own IPFS nodes or IPFS clusters to become a part of the entire IPFS network of the system. Encryption components in the system including ABE and MA-ABE are implemented to support privacy and data access rights. In each CA, end-user services are created to provide authentication, file reading and writing, and cryptography. Note that all IPFS nodes are connected to each other and each service also plays a role as an end point of the blockchain network.

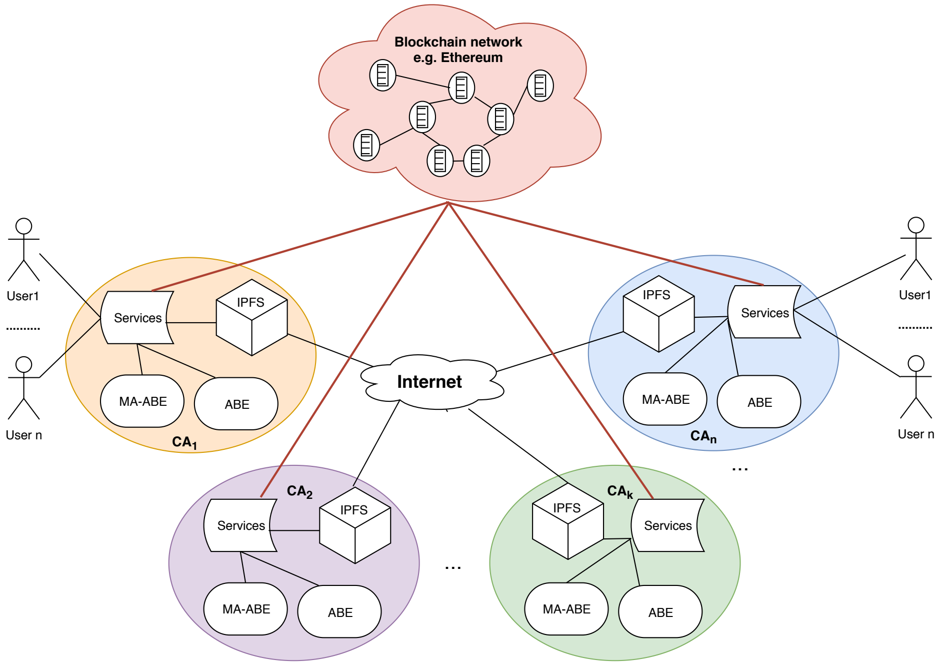

Fig. 1. Overall architecture

The flow of encrypted data could be described as follows. The encrypted data will be pushed to IPFS, and only users with policy-compliant properties will be able to decrypt and access the plain text of that content. Similarly, in the case of content sharing between users of different organizations, our system could apply both ABE and MA-ABE mechanisms. For example, the owner data will use MA-ABE to encrypt data with the properties of the organizations that the user wants to share. Only organizations with properties that satisfy the policy can decrypt and access the plain text of that data.

Each organization will deploy Services components with the purpose of supporting users to easily connect and use the services of component IPFS, ABE and MA-ABE respectively. In addition, services also support the conversion of encryption methods, deployment of encryption services, and integrated storage in accordance with the requirements and business of the organization. In this design, the services are centralized and easy to be attacked. However, there are multiple services that could be deployed to ensure the fault tolerance characteristic.

The purpose of using a blockchain network is to apply to business, and requirements that need to ensure the transparency, immutability and integrity of data. Ethereum is an example of a public blockchain network that can be employed in our design. Ethereum has an amazing ecosystem with Ethereum network, cryptocurrencies (Ether), Smart Contract, etc. and is especially easy to use. With the use of IPFS for storage makes our system easy to expand when the number of users and resources increases. Hence, our system can achieve the highly elasticity.

Due to high fault tolerance, any attack on one organization will not affect other organizations, all data will always be safe because it has been backed up by another organization. The problem of managing and sharing secret content resources between users inside and outside the organization are solved by ABE and MA-ABE. The system is highly transparent. By providing services in the form of web interfaces, users do not need to care about the complex architecture of the system, interaction using the services is easy and comfortable.

Table I shows the comparison between B-Box with several current systems listed in the related work. B-Box is the only one that fully meets the requirements of decentralization, data security using ABE and MA-ABE as well as the immutability and transparency of data.

V. Example Use Cases

We present two example use cases to illustrate processes of data sharing. In the first use case, an organization needs to create (PK, MSK) for ABE encryption. In the latter use case, a generation of (PK, SK) is necessary to conduct MA-ABE encryption. In addition, MA-ABE requires an identifier of GID for each user that is a hash of sha256 from user's email.

A. File Sharing in an Organization

Data sharing process in an organization using ABE is described in Figure 2. In this figure, we illustrate the flow of sharing and managing data access rights of three users including data owner, User1, User2 in the same organization. Assume that the attributes of User1 and User2 are $user1.Attr$ and $user2.Attr$, respectively. When the owner wants to share data with User1 and User2, he/she will set up a policy p based on attributes of User1 and User2 as follows:

p = $user1.Attr$ OR $user2.Attr$

TABLE I
A COMPARISON BETWEEN B-BOX AND RELATED SYSTEMS

	Decentralized	Secure with ABE	Secure with MA-ABE	Transparent
Zheng et al. [27]	✓	✗	✗	✓
Xu et al. [28]	✓	✗	✗	✓
Nizamuddin et al. [16]	✓	✗	✗	✓
Wang et al. [3]	✓	✓	✗	✓
B-box	✓	✓	✓	✓

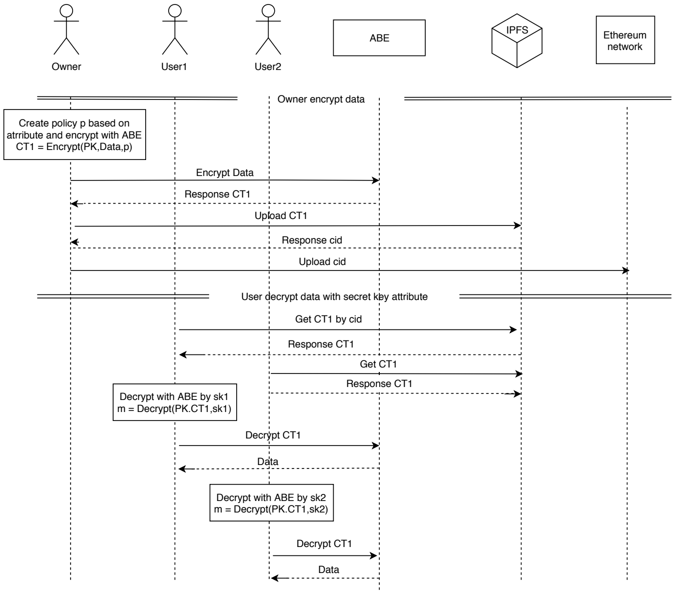
Owner
User1
User2
ABE
IPFS
Ethereum network
Owner encrypt data
Create policy p based on atrribute and encrypt with ABE CT1 = Encrypt(PK,Data,p)
Encrypt Data
Response CT1
Upload CT1
Response cid
Upload cid
User decrypt data with secret key attribute
Get CT1 by cid
Response CT1
Get CT1
Response CT1
Decrypt with ABE by sk1 m = Decrypt(PK.CT1,sk1)
Decrypt CT1
Data
Decrypt with ABE by sk2 m = Decrypt(PK.CT1,sk2)
Decrypt CT1
Data

Fig. 2. Process of file sharing among users in an organization

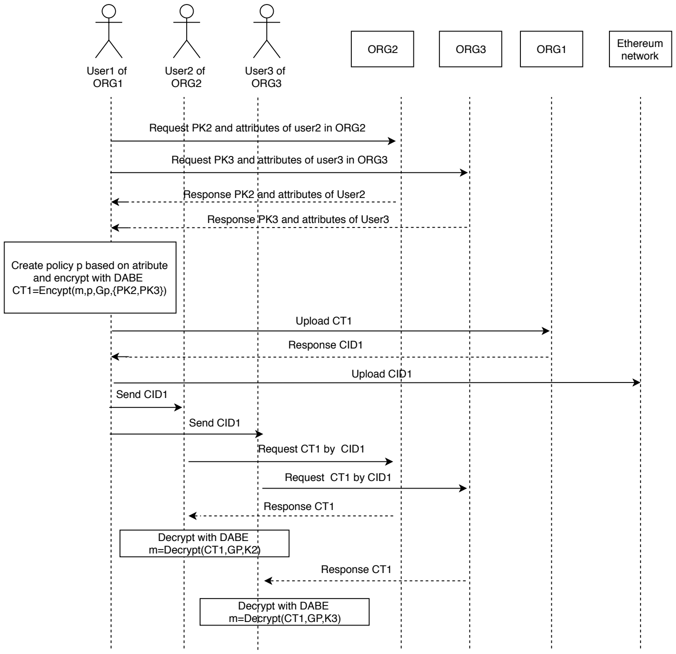
User1 of ORG1
User2 of ORG2
User3 of ORG3
ORG2
ORG3
ORG1
Ethereum network
Request PK2 and attributes of user2 in ORG2
Request PK3 and attributes of user3 in ORG3
Response PK2 and attributes of User2
Response PK3 and attributes of User3
Create policy p based on atribute and encrypt with DABE CT1=Encypt(m,p,Gp,{PK2,PK3})
Upload CT1
Response CID1
Upload CID1
Send CID1
Send CID1
Request CT1 by CID1
Request CT1 by CID1
Response CT1
Decrypt with DABE m=Decrypt(CT1,GP,K2)
Response CT1
Decrypt with DABE m=Decrypt(CT1,GP,K3)

Fig. 3. Process of file sharing among users in multiple organizations

After that, ciphertext of data is created by using the Encrypt algorithms of ABE. Ciphertext will be uploaded to the IPFS network, and then the user can receive CID of the ciphertext. To ensure the transparency and immutability of the data, the owner by using Services components will push the CID into Ethereum network. Other users can get the CID from Ethereum network, download the ciphertext from IPFS through this CID, and easily decrypt to obtain original data by using Decrypt algorithms of ABE if they have attributes that are used to built policy p. Users who do not have such attributes cannot access to the plain text of data. With ABE, instead of having to encrypt multiple times for the same plain text to share with multiple people, a user only needs to encrypt once. Along with that, a single ciphertext helps optimize the storage resources.

B. File Sharing in Multiple Organizations

Figure 3 describes the process for sharing data using MA-ABE among users of different organizations. In this figure, we illustrate the flow of sharing and managing data access rights of three users including User1, User2, User3 belonging to three organizations ORG1, ORG2, ORG3, respectively. When User1 wants to share data with User2 and User3, he/she needs to send a request to organizations ORG2 and ORG3 to retrieve public keys and attributes related to User2 and User3. Assume that the attributes of User2 and User3 are $user2.Attr$ and $user3.Attr$, respectively. After receiving these attributes, User1 makes the use of them to build a policy p that allows User2 and User3 to decrypt the ciphertext as follows.

p = $user2.Attr$ OR $user3.Attr$

Next, User1 encrypts data by using Encrypt algorithms of MA-ABE, which is presented in Section II. Ciphertext CT1 then will be uploaded to the IPFS network and User1 will receive CID1. CID1 will be pushed to Ethereum, which is similar to the process of sharing data inside an organization. Everyone can get CID1 from Ethereum network and use it to get the ciphertext CT1 from IPFS network. However only User1, User2 and User3 can decrypt it to get the plain text by using Decrypt algorithms and their attributes.

With MA-ABE, the data is only encrypted once, and the system only needs to store a single ciphertext on IPFS while users can decrypt using different private keys. MA-ABE also guarantees the confidentiality when only users who own the private keys associated with the policy satisfaction attribute can decrypt the ciphertext. Users who do not have the key that satisfies the policy cannot decrypt the ciphertext. So we achieve fine-grained access control through encryption data.

VI. Conclusion and Future Work

In this paper, we aim to develop a decentralized, secure, and transparent storage system. To this end, we first analyze related approaches for decentralized storage, data encryption, and make them trustworthy. We then develop a storage system, which relies on IPFS network, ABE and MA-ABE encryption, and Ethereum blockchain. We demonstrate two use-cases to show advantages of our system in establishing trust and feasibility for users. At present, our system was implemented, and providing preliminary results. As a next step, we will focus on system evaluation of performance and privacy mechanism. In addition, we also plan to analyze and apply recent researches on encryption technologies to enhance security aspect of this system.

Acknowledgment

This work is supported by the Vietnam national project ”Research on building an online public service in the field of land management using blockchain technology” under project number KC.01.27/16-20.

References

[1] G. Wang, Q. Liu, and J. Wu, “Hierarchical attribute-based encryption for fine-grained access control in cloud storage services,” in *Conference on Computer and communications security*, 2010, pp. 735–737.

[2] S. S. Iropia and R. Vijayalakshmi, “Decentralized Access Control of Data Stored in Cloud using key Policy Attribute based Encryption,” *International Journal of Inventions in Computer Science and Engineering*, vol. 1, 2014.

[3] 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. 38 437–38 450, 2018.

[4] J. Isaak and M. J. Hanna, “User data privacy: Facebook, cambridge analytica, and privacy protection,” *Computer*, vol. 51, no. 8, pp. 56–59, 2018.

[5] C. Cadwalladr and E. Graham-Harrison, “Revealed: 50 million facebook profiles harvested for cambridge analytica in major data breach,” *The Guardian*, vol. 17, p. 22, 2018.

[6] J. Benet, “IPFS - Content Addressed, Versioned, P2p File System,” [Online; accessed 01 Nov. 2019]. Available: https://www.hirego.io/.

[7] Y. Peng, W. Zhao, F. Xie, Z.-h. Dai, Y. Gao, and D.-q. Chen, “Secure cloud storage based on cryptographic techniques,” *Journal of China Universities of Posts and Telecommunications*, vol. 19, pp. 182–189, 2012.

[8] S. Zhu, X. Yang, and X. Wu, “Secure Cloud File System with Attribute Based Encryption,” in *International Conference on Intelligent Networking and Collaborative Systems*, 2013, pp. 99–102.

[9] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Conference on Computer and communications security*, 2006, pp. 89–98.

[10] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in *IEEE symposium on security and privacy*, 2007, pp. 321–334.

[11] A. Lewko and B. Waters, “Decentralizing Attribute-Based Encryption,” in *Advances in Cryptology*, D. Hutchison, T. Kanade, J. Kittler, J. M. Kleinberg, F. Mattern, J. C. Mitchell, M. Naor, O. Nierstrasz, C. Pandu Rangan, B. Steffen, M. Sudan, D. Terzopoulos, D. Tygar, M. Y. Vardi, G. Weikum, and K. G. Paterson, Eds., 2011, vol. 6632, pp. 568–588.

[12] S. Nakamoto, “Bitcoin: A peer–to–peer electronic cash system,” [Online; accessed 01 Nov. 2019]. Available: https://bitcoin.org/bitcoin.pdf.

[13] Z. Zheng, S. Xie, H. Dai, X. Chen, and H. Wang, “An overview of blockchain technology: Architecture, consensus, and future trends,” in *IEEE International Congress on Big Data*, 2017, pp. 557–564.

[14] S. Seebacher and R. Schüritz, “Blockchain technology as an enabler of service systems: A structured literature review,” in *International Conference on Exploring Services Science*, 2017, pp. 12–23.

[15] J. L. Zhao, S. Fan, and J. Yan, “Overview of business innovations and research opportunities in blockchain and introduction to the special issue,” *IEEE Access*, vol. 6, pp. 2169–3536, 2018.

[16] N. Nizamuddin, K. Salah, M. Ajmal Azad, J. Arshad, and M. Rehman, “Decentralized document version control using ethereum blockchain and IPFS,” *Computers & Electrical Engineering*, vol. 76, pp. 183–197, 2019.

[17] T. C. Dao, B. M. Nguyen, and B. L. Do, “Challenges and strategies for developing decentralized applications based on blockchain technology,” in *International Conference on Advanced Information Networking and Applications*, 2019, pp. 952–962.

[18] X. Xu, I. Weber, M. Staples, L. Zhu, J. Bosch, L. Bass, C. Pautasso, and P. Rimba, “A taxonomy of blockchain-based systems for architecture design,” in *International Conference on Software Architecture (ICSA)*, 2017, pp. 243–252.

[19] M. Steichen, B. Fiz, R. Norvill, W. Shbair, and R. State, “Blockchain-Based, Decentralized Access Control for IPFS,” in *International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 1499–1506.

[20] M. Vukolić, “The quest for scalable blockchain fabric: Proof-of-work vs. bft replication,” in *International workshop on open problems in network security*, 2015, pp. 112–125.

[21] A. Baliga, “Understanding blockchain consensus models,” in *Persistent*, 2017.

[22] Y. Hirai, “Defining the ethereum virtual machine for interactive theorem provers,” in *International Conference on Financial Cryptography and Data Security*, 2017, pp. 520–535.

[23] P. H´ector, Sanju´and Samuli and T. Pedro, “Merkle-crdts (draft),” [Online; accessed 01 Nov. 2019]. Available: https://docs.ipfs.io/guides/concepts/merkle-dag/.

[24] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in *International Workshop on Public Key Cryptography*, 2011, pp. 53–70.

[25] A. Lewko and B. Waters, “Decentralizing attribute-based encryption,” in *Annual international conference on the theory and applications of cryptographic techniques*, 2011, pp. 568–588.

[26] R. Norvill, B. B. Fiz Pontiveros, R. State, and A. Cullen, “IPFS for Reduction of Chain Size in Ethereum,” in *International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, 2018, pp. 1121–1128.

[27] Q. Zheng, Y. Li, P. Chen, and X. Dong, “An Innovative IPFS-Based Storage Model for Blockchain,” in *International Conference on Web Intelligence (WI)*, 2018, pp. 704–708.

[28] Q. Xu, Z. Song, R. S. Mong Goh, and Y. Li, “Building an Ethereum and IPFS-Based Decentralized Social Network System,” in *International Conference on Parallel and Distributed Systems (ICPADS)*, 2018, pp. 1–6.

[29] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Conference on Computer and communications security*, 2006, pp. 89–98.

[30] S. Zarandioon, D. Yao, and V. Ganapathy, *K2C: Cryptographic Cloud Storage with Lazy Revocation and Anonymous Access*, M. Rajarajan, F. Piper, H. Wang, and G. Kesidis, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, vol. 96.

[31] S. J. De and S. Ruj, “Decentralized Access Control on Data in the Cloud with Fast Encryption and Outsourced Decryption,” in *Global Communications Conference*, 2015, pp. 1–6.