OPTIMZIED HYPERLEDGER FABRIC BLOCKCHAIN POLICY-BASED BROADCAST ACCESS AUTHORIZATION (OHFB-PBAA) SYSTEM FOR SECURED DATA SHARING IN CLOUD COMPUTING

C. Nagarani, C. Thirumoorthi Assistant Professor Department of Computer Science PSG College of Arts & Science Coimbatore. ranipsg22@gmail.com

Abstract

Cloud computing's expandability and on-demand deployment features have made it a generally accepted alternative for data upkeep. Serious worries regarding the privacy of data in the cloud persist despite the conveniences and advantages that cloud computing offers. But the majority of systems only allow coarse-grained or single-recipient re-encryption, which might reduce the flexibility for sharing data. The well-known Identity-Based Broadcast Encryption (IBBE) and key-policy attribute-based encryption are introduced into PRE to create the Policy-based Broadcast Access Authorization (PBAA) scheme, which addresses this problem. These methods still have issues with security and communication overhead for data users, and they do not completely eliminate the single point of failure. The authors suggest utilizing blockchain technology and the Cipher text policy Attribute-Based Encryption (CP-ABE) algorithm to create the Optimized Hyperledger Fabric Blockchain Policy-Based Broadcast Access Authorization (OHFB-PBAA) solution, which secures end users' data against cloud threats.According to the Cipher text policy scheme, OHFB is able to comprehend both the secure distribution of user attribute keys in the blockchain data attribute encryption scheme and the user-level fine-grained security access to control blockchain data. Furthermore, transactions can be automatically processed by the blockchain's smart contract, which drastically cuts down on the scheme's running time and security. The findings demonstrate that, without imposing appreciable computational costs on cloud servers or data consumers, the suggested OHFB-PBAA method accomplishes an effective and adaptable access authorization for data sharing.

Keyword:

Cloud computing, Proxy re-encryption, Broadcast encryption, Policy-Based Broadcast Access Authorization (OHFB-PBAA), Hyperledger Fabric Blockchain (HFB), Cipher text policy Attribute-Based Encryption (CP-ABE), and security.

1. INTRODUCTION

Since cloud computing may be expanded and can be deployed on-demand, it has gained widespread acceptance as a data maintenance option[1], [2]. It is a developed storage platform and has many advantages including low cost and scalability [3,4,5]. Therefore lot of businesses and people frequently outsource their data to the cloud so that it can be stored there and shared with authorized users upon request.. For example, In a cloud-based health information system, patients upload data about their health to the cloud so that medical expert can use it to diagnose conditions. Third-party cloud service providers such as Microsoft Health Vault [6] and Google Health [7], has makes sharing of the health information among different medical institutions or individuals more convenient and efficient. Similarly, Enterprise managers desire to offer their authorized employees with access to large data wherever it is needed, in addition to storing it in the cloud.

In addition to saving local storage space, outsourcing data for cloud sharing also significantly lowers the cost of software acquisition and hardware upkeep for businesses[8,9,10]. Although people take advantages of this new technology and service, their concerns about data security arises as well. There are significant concerns over the privacy of data on the cloud, notwithstanding the ease and advantages that cloud computing offers. The standard approach is to first encrypt the data before sending it to the cloud. However, sharing data with different users becomes more challenging with such a method. The data owner can, of course, download the encrypted text from the cloud, decrypt it with his private key, and then encrypt the data for each recipient in turn. This straightforward method, nevertheless, is too complicated and inefficient. Proxy re-encryption (PRE) could be useful to address the dilemma for sharing encrypted data [11]. All the benefits of PRE are retained by Identity-based Proxy re-encryption (IBPRE), which also reduces the effort of managing public keys in conventional PRE systems by enabling any recognized string to be used as a public key.[12]. To further illustrate this point, consider the following scenario.Suppose that Alice can access the encrypted genome data uploaded by different volunteers. Before uploading, volunteers have labelled the genome data with descriptive tags, e.g., a volunteer uses a set of tags f"Female", "Age"=30, "Diabetes's to indicate that the genome data belongs to a 30-year-old woman who has the inherited diabetes. For collaborative work with a group of peers, Alice would like to share some genome data with the peers. For instance, Alice wants to share the genome data of 20-40 years old women who may have diabetes or heart disease, i.e., the data whose tags meet a policy: "Female" AND "Age" [20; 40] AND ("Diabetes" OR "Heart Disease") (See Fig. 1). Since the genome data have already been encrypted, the peers cannot directly access them. Therefore, In order to transform the encrypted genetic material that complies with the access policy into cipher texts that the group of peers can decrypt, Alice could require a flexible re-encryption process.

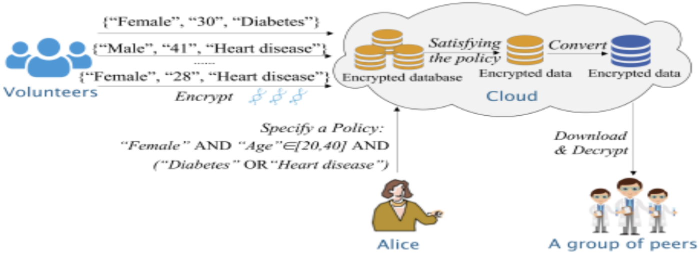

Fig. 1 Flexible Data Sharing in the Cloud

IBPRE's single recipient and "all-or-nothing" data exchange restrictions make it difficult for use in cloud environments. identity-based conditional Proxy re-encryption (IBCPRE [13]), which allows Alice to set a condition for a re-encryption key so that the proxy can only convert the encrypted data matching the condition, was presented by Shao et al. to address the "all-or-nothing" problem.Since IBCPRE prohibits the specification of numerous conditions in a re-encryption key, Alice must still create a large number of re-encryption keys in the event that she needs to specify several criteria for data sharing. Furthermore, a number of users cannot view the data simultaneously because IBCPRE only allows single-recipient data sharing.

While several issues with the cloud have been resolved by current cryptography-related technologies, the single point of failure issue remains unresolved. However, medical data management also uses a centralized approach called Policy-based Broadcast Access Authorization (PBAA), which is typically based on Broadcast Access [14]. Unfortunately, in order to limit the accessibility of various data consumers, the PBAA architecture necessitates implementing intricate policies. Because the data is stored centrally, there is a particular vulnerability in the process of configuring and changing the policy and cipher text format. An attacker might use this vulnerability to elevate privileges and get the grant for the full dataset.

Fan et al. proposed a fine-grained access-control scheme based on Ciphertext-Policy Attribute-Based Encryption (CP-ABE) and Trusted Execution Environment (TEE) [15]. TEE is employed as a trusted computing environment to protect encrypted data [16, 15]. Blockchain technology is used to ensure the integrity of encrypted data. In the work by Jemel and Serhrouchni [17] proposed a dynamic access control strategy based on blockchain technology and CP-ABE. The time attribute is introduced to realize the dynamic access of data, and only a user whose attribute meets the access control policy within the specified time can access the data. A novel technique called blockchain technology can successfully address the issues with conventional PBAA control. Blockchain technology is distinguished from traditional centralized storage architecture by its decentralization and nontamperability, which may efficiently ensure data security and enhance system scalability. Blockchain thus offers a viable replacement for centralized cloud storage. A decentralized way for participants from various organizations to authenticate requests for access to data is made possible by smart contracts built on blockchain technology.

In this paper, investigate how to achieve secure and flexible data sharing in cloud computing. For the first time, novel notion called an Optimized Hyperledger Fabric Blockchain Policy-Based Broadcast Access Authorization (OHFB-PBAA) approach by integrating the Key policy Attribute-Based Encryption (KP-ABE) algorithm and blockchain technology. Also define the notion (including scheme definition and security model) of OHFB-PBAA and instantiate a concrete OHFB-PBAA scheme.

Data Access Control Mechanism: This technique is appropriate for the Hyperledger Fabric's data access control mechanism. The blockchain network of ledgers achieves the highest level of user-based granularity data access control while ensuring that data is not leaked. Although cloud storage has low efficiency, several researchers have realized that blockchain cannot store vast amounts of data. This study proposes an access control strategy that guarantees control process simplicity while implementing access control using blockchain and CP-ABE. Consequently, the plan achieves the ideal mix between effective cloud file sharing and fine-grained access control.

Policy-based Re-encryption: In the event that the data owner decides to distribute encrypted data in the broadcast ciphertexts to a new user group, they can create an access policy that will produce a delegation key. With this key, any initial ciphertext that satisfies the access policy can be transformed by the cloud into a new ciphertext that allows the new user group to access the underlying data.

Data Privacy Protection: The PBAA system provides strong protection for the privacy of outsourced data. To be more precise, no user can access encrypted data if they do not have the correct private key. Moreover, even if the cloud has the delegation key, it is unable to access encrypted data or determine a valid delegation key to re-encrypt the ciphertext if the data owner has not provided one.

2. LITERATURE SURVEY

Deng et al [14] proposed a Policy-based Broadcast Access Authorization (PBAA) scheme by introducing the well-established identity-based broadcast encryption (IBBE) and key-policy attribute-based encryption into PRE. In PBAA scheme, a data owner can apply IBBE to encrypt his data to a group of recipients. More importantly, the data owner can generate a delegation key with an access policy, and send this key to the cloud such that it can convert any initial ciphertext satisfying the access policy into a new ciphertext for a new group of recipients. With these features, cloud users can share their remote data in a secure and flexible way. Security analysis and performance evaluation show that the PBAA scheme is secure and efficient, respectively.

Lai et al. [18] constructed an anonymous identity-based broadcast encryption (IBBE), which offers the user revocation of ciphertext and the revocation process does not reveal any information of the plaintext and receiver identity. In proposed scheme, the group of receiver identities is anonymous and only known by the encryptor. Lai et al. [19] presented Fully Privacy-Preserving and Revocable IBBE, which preserves the data privacy and the identity privacy of the receiver as well as the revoked user. The security of proposed scheme is proved to be semantically secure in the random oracle model. Moreover, Attribute-Based Encryption (ABE) provides a finegrained access control on data [20], i.e., a data owner can specify an access policy to encrypt data so that only the users whose attributes satisfy the access policy can access the data.

Ge et al. [21] proposed an identity-based broadcast PRE scheme which can convert a ciphertext for a set of recipients into a new ciphertext for a new set of recipients. Deng et al. [22] proposed a new paradigm called hybrid attribute-based proxy re-encryption (HAPRE). In HAPRE, a semitrusted proxy can be authorized to convert ciphertexts of an ABE scheme into ciphertexts of an identity-based encryption (IBE) scheme without letting the proxy know the underlying messages. With this technique, the data owner generates a delegation key by first encrypting a secret value via the IBBE encryption and then blinding his private key with the secret value under an access policy; in the decryption, data users first obtain the secret value and then apply it to recover the plaintext.

Recently, Deng et al. [23] also proposed an IBPRE that can convert a single-recipient ciphertext into a multi-recipient broadcast ciphertext. Yin et al. [24] proposed a broadcast IBCPRE scheme while a ciphertext can be shared only once. This system reduces the consumption of system, but realizes the encryption efficiency and security. Huang et al. [25] proposed a PRECISE, an identity-based private data sharing scheme in Online Social Networks(OSNs) with big data, in which the data owner could broadcast private data to a group of users at one time in a convenient and secure way. In order to achieve secure and fine-grained data disseminating in OSNs, attribute-based conditional proxy re-encryption is used to guarantee that only the data disseminators whose attributes satisfy access policy can disseminate the data to their own social space. The theoretical analysis and experimental results prove the security and efficiency.

Ge et al. [26] proposed a fine-grained identity-based proxy broadcast re-encryption scheme to support multi-recipient data sharing. It has been empirically analyzed, while, unfortunately, the security cannot be formally to be verified, since neither threat model nor mathematical proof is given to cloud. Kim et al. [27] presented an adaptively secure identity-based broadcast encryption system featuring constant sized ciphertext in the standard model. Dual system encryption technique is proposed which offers adaptive security under the general subgroup decisional assumption. Traditional broadcast encryption [27] relies on a third party to manage public-key certificates of all users, which would incur a singlepoint problem. Proposed scheme demonstrates that the adaptive security of the schemes utilizing a compositing order group can be proven under the general subgroup decisional assumption. However, there is a

major challenge in applying the above techniques to construct the PBAA scheme. That is, the data owner is hard to realize the access policy in the delegation key without knowing the system master secret key. Specifically, to generate a functional delegation key, the data owner should split the master secret key into shares and assign each share to a condition involved in the access policy. It is unable for the data owner to do this without having the master secret key. To overcome this challenge, a novel approach is presented for applying the Optimized Hyperledger Fabric Blockchain (OHFB) to split the data owner's private key according to the access policy. Since the master secret key is properly embedded in the private key, the shares of the secret can still be obtained. These shares then can be used in the re-encryption to convert the ciphertexts that satisfy the access policy

Qin et al [28] proposed a Blockchain-based Multi-authority Access Control scheme (BMAC) for sharing data securely. Shamir secret sharing scheme and permissioned blockchain (Hyperledger Fabric) are introduced to implement that each attribute is jointly managed by multiple authorities to avoid single point of failure. Moreover, blockchain helps to record the access control process in a secure and auditable way. Finally, analyze the security of the proposed algorithm. Further analysis and comparison show the performance of the proposed method.

Gao et al [29] combined blockchain, ciphertext-policy attribute-based encryption (CP-ABE), and InterPlanetary File System (IPFS) to address this problem to propose a blockchain-based security sharing scheme for personal data (BSSPD). In this usercentric scheme, the data owner encrypts the sharing data and stores it on IPFS, which maximizes the scheme's decentralization. The address and the decryption key of the shared data will be encrypted with CP-ABE according to the specific access policy, and the data owner uses blockchain to publish his data-related information and distribute keys for data users. Thorough analysis of the storage and computing overhead proved that BSSPD has a good performance than existing methods.

Eltayieb et al [30] combined the concept of blockchain with attribute-based signcryption to provide a secure data sharing in the cloud environment. Smart contract solves the problem of cloud storage such as returning wrong results as in the traditional cloud server. The proposed scheme satisfies the security requirements of the cloud computing such as confidentiality and unforgeability. Alniamy and Taylor [31] proposed system is implemented by combining Hyperledger blockchain technology and Attribute-based Encryption (ABE) scheme to achieve this fine-grained access control of the shared files in decentralized environment. Proposed system prototype was implemented using chaincodes and tested on the Hyperledger Composer blockchain platform.

Deb et al. [32] proposed scheme can realize the user-level fine-grained security access to control blockchain data while also realizing the secure distribution of user attribute keys in the blockchain data attribute encryption scheme based on the ciphertext policy scheme. Hyperledger Blockchain networks also the security objectives of secure transmission of user characteristic secret keys and data privacy protection. The performance analysis part also shows that the proposed scheme has good usability.

2. PROPOSED METHODOLOGY

The Cipertext policy Attribute-Based Encryption (CP-ABE) algorithm and blockchain technology are integrated in the Optimized Hyperledger Fabric Blockchain Policy-Based Broadcast Access Authorization (OHFB-PBAA) strategy to protect end users' data from cloud threats. Depending on the Cipertext policy scheme, OHFB is able to comprehend user-level fine-grained security access control over blockchain data while also realizing the secure distribution of user attribute keys in the blockchain data attribute encryption scheme. Furthermore, transactions can be automatically processed by the blockchain's smart contract, which drastically cuts down on the scheme's running time and security.

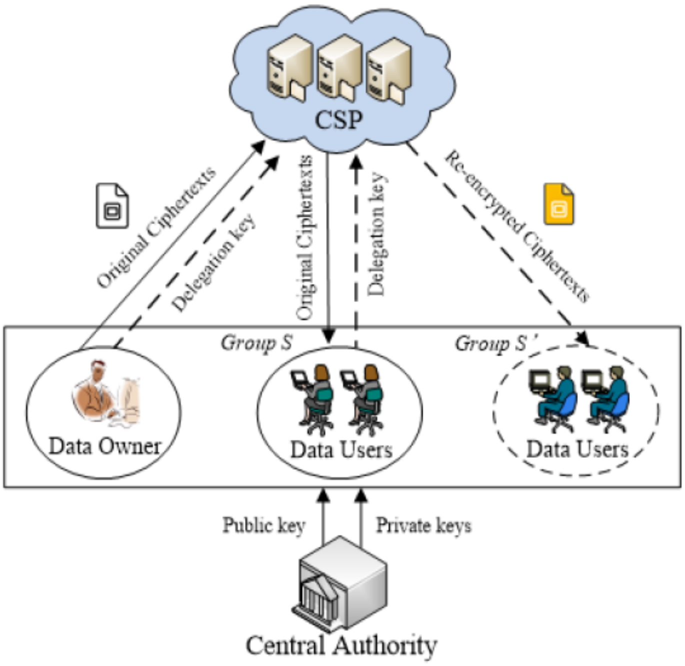

Fig. 2 System Architecture of OHFB-PBAA Schema

The OHFB-PBAA system consists of four entities (as shown in Fig. 2),that is, a central authority (CA), cloud service provider (CSP),data owners and data users. CA is a fully trusted party responsiblefor publishing system public key and responding to registrationrequests from data owners and data users. CSP has abundant resourcesto provide storage and computation services. Specifically,CSP provides storage service for data owners to store the ciphertextsof their data, and provides computing service to transformthe ciphertexts into re-encrypted ones. Thus, CSP stores bothoriginal ciphertexts and re-encrypted ciphertexts. In real-worldscenarios, an organization can buy the storage and computingservices provided by CSP, and the IT center of the organizationplays the role of CA. Then all the employees registered at CA canmake use of the cloud services.

Data owners can outsource their data to CSP for data sharing.Specifically, a data owner can designate a set S of identities of theintended data users and a set L of descriptive conditions (e.g.,keywords of the data content), encrypting his data with thesetwo sets, and then outsources the resulting (original) ciphertextto CSP. When data users designated in the set S are online, theycan retrieve the ciphertext from CSP and decrypt it using theirprivate keys. If the data owner or an authorized data user in Swants to share some data to a new group (denoted by $S^{'}$) of users,the data owner formulates an access structure A over conditionsand then generates a delegation key (i.e., re-encryption key) withA and the identities of the users in $S^{'}$. With such delegation key,CSP converts the original ciphertexts with conditions satisfyingthe access structure into the re-encrypted ciphertexts such that allthe users in $S^{'}$ can decrypt using their own private keys.Considering these realistic attacks, identify the security goals of PBAA as follows.

1.1 PBAA SECHEMA

Privacy of outsourced data: Access to encrypted data is restricted to users who possess the appropriate private keys, in the event that the data is outsourced to CSP. Furthermore, the data users who possess the appropriate private keys are the only ones who can access the re-encrypted data. Put differently, CSP and unauthorized data users (who lack the proper private keys) cannot decrypt any encrypted data.

Specificity of delegation key: The ciphertexts with criteria that satisfy the access structure are the only ones that can be converted using a delegation key that has been issued by a data owner. In order to re-encrypt nonspecific ciphertexts, neither the CSP nor permitted data users can misuse the delegation key nor figure out a legitimate delegation key.

Seven algorithms PBAA systems uses Seven algorithms : Dec1 and Dec2, DKGen, ReEnc, Setup, Register, Encrypt, and Dec2. The master secret key MSK and system public key PK are generated at startup by the CA using the Setup procedure. It maintains MSK confidential while disclosing PK to third parties. A user should always seek CA for registration whenever they want to join the system. When a person is approved to participate, CA first establishes a distinct identity ID, after which it calls the Register procedure to build a privacy key using ID. To safeguard data privacy, the data owner can use the encryption algorithm prior to outsourcing data to the cloud.To be more precise, the data owner encrypts the data using S and L after first defining the set S of receivers' identities and the set L of descriptive conditions. The generated ciphertext CT is then outsourced to CSP by the data owner. The only things the data owner needs to do to share encrypted data with a new group of recipients are figure out who the new recipients are and create an access policy that outlines the kind of data the owner wants to share. Subsequently, the data owner uses S' and the DKGen technique to generate a delegation key. By using the delegation key, CSP calls the ReEnc algorithm to re-encrypt any ciphertext with set L satisfying A into a new ciphertext called CT'.A data user can download a ciphertext from CSP and attempt to decrypt it once he is online. If the data user's identity is supplied in the set S (resp. S'), he calls the Dec1 (resp. Dec2) algorithm to retrieve the underlying data for an original ciphertext CT (resp. a re-encrypted ciphertext CT').

The PBAA scheme is built on bilinear groups. Suppose G and G$_T$ are two finite cyclic groups of prime order p. Let g be a generator of G. The group G is a bilinear group if there exists anefficient bilinear map $e : G \times G \rightarrow G_T$ such that: i) for all $u, v \in G$ and a; $b \in Z_p$, $e(u^a, v^b) = e(u, v)^{ab}$ and ii) e(g; g)≠1

Access structure: In the PBAA scheme, the access policyassociated with a delegation key will be represented as an accessstructure. Let $\{P_1, \dots, P_n\}$ be a set of parties. A collection$\mathbb{A} \subseteq 2^{\{P_1, \dots, P_n\}}$ is monotone if for $\forall B, C$, have that $C \in \mathbb{A}$holds if $B \in \mathbb{A}$ and $B \subseteq C$. An access structure (respectively,monotone access structure) is a collection (respectively, monotonecollection) $\mathbb{A}$ of non-empty subsets of $\{P_1, \dots, P_n\}$, i.e., $A \subseteq 2^{\{P_1, \dots, P_n\}}\{\emptyset\}$. The sets in $\mathbb{A}$ are called the authorized sets,and the sets not in $\mathbb{A}$ are called the unauthorized sets.

Linear Secret Sharing Scheme (LSSS): To realize the access structure, employ theLSSS in the delegation-key generation. A secret-sharing scheme∏ over a set of parties $\mathbb{P}$ is called linear (over $\mathbb{Z}_p$) if: i) the sharesfor each party form a vector over $\mathbb{Z}_p$; ii) there exists a matrix Acalled the share-generating matrix for ∏, where A has l rows andq columns. Given an LSSS ($\mathbb{A}$; ρ) for access structure $\mathbb{A}$ and anauthorized set $L \in \mathbb{A}$, there must exist constants $\{\omega_i \in \mathbb{Z}_p\}$ thatcan help to recover the secret δ.

System Setup: In this stage, CA initializes the PBAA scheme by calling theSetup algorithm. Suppose that a company buys cloud services forits employees to store and share business data. Then the IT centerof the company can play the role of CA to initialize the PBAAscheme so that all the employees can apply the PBAA schemeto protect the privacy of the business data. By running Setupalgorithm, CA generates system public key PK and master secretkey MSK.

Registration: In the registration stage, CA first checks whether a user is allowedto join in the system. For example, the IT center (CA) of acompany checks whether a user requesting to use the cloudservices is a valid employee. If yes, CA generates a private key andsends it to the user as an authorized credential to access the datastored in the cloud. To generate a private key, CA first determinesa unique identity ID for the user

Encryption: With the PBAA scheme, a data owner can securely share datawith a group of recipients. Specifically, the data owner firstdetermines a set S of identities of the recipients. Assumethat $S = \{ID_i\}_{1 \le i \le N}$ in which ID_i denotes the identity ofthe i^{th} recipient. In theapplication of the PBAA scheme, a data owner first picks a randomkey $M \in G_T$ and then encrypts M with the sets S and L bycalling the following Encrypt algorithm.

$CT \leftarrow Encrypt(PK, M, S, L)$: Given the system public key PK, the set S of identities, the set L of conditions, and the message $M \in G_T$to be encrypted, the data owner first chooses a random $s \in \mathbb{Z}_p$.

Delegation: After a volume of data have been encrypted and outsourced to the cloud, the data owner (or authorized user in S) can still share some data with a new group of recipients. Suppose that the data owner wants to share the data regarding the turnover and development cost of an electronic product in the first quarter. The data owner first choosesrandom values $\delta; s' \in \mathbb{Z}_p$ and computes

$$d_0 = g^{\delta}.F\left(e(g,\mu)^{s'}\right), d_1 = g^{s' \prod_{ID_i' \in S'}\left(\alpha + H(ID_i')\right)}, d_2 = \mu_1^{-s'} \quad (1)$$

Re-Encryption: Upon receiving the delegation key $DK_{ID \to S'|\mathbb{A}}$ from the data owner, CSP applies it to transform the original ciphertexts to be accessible to a new set $S^{'}$ of recipients. CSP first searches the original ciphertexts generated under S and L such that $ID \in S$ and $L \in \mathbb{A}$. Here, the condition ID 2 S implies that the delegation key can only be used to re-encrypt the data owner's ciphertexts, and the condition $L \in \mathbb{A}$ means that only a subset of ciphertexts, specified by the data owner, can be re-encrypted. After finding all the ciphertexts satisfying the two conditions, CSP uses $DK_{ID \to S'|\mathbb{A}}$ to re-encrypt each ciphertext.

Decryption: When a data user gets online, he can download a ciphertext fromCSP and try to decrypt it using his private key. Note thatthere are two kinds of ciphertexts stored in the cloud, i.e., originalciphertexts and re-encrypted ciphertexts.

3.2 OHFB-PBAA approach

In order to enhance the security of key-policy attribute-based encryption into PRE, here Cipertext policy Attribute-Based Encryption (CP-ABE) algorithm and blockchain technology is introduced to data storage. The architecture of the data access control scheme that combines CP-ABE and blockchain technology consists of four layers, namely, the consumption layer, interaction layer, access control layer, data layer, as shown in Fig. 3.

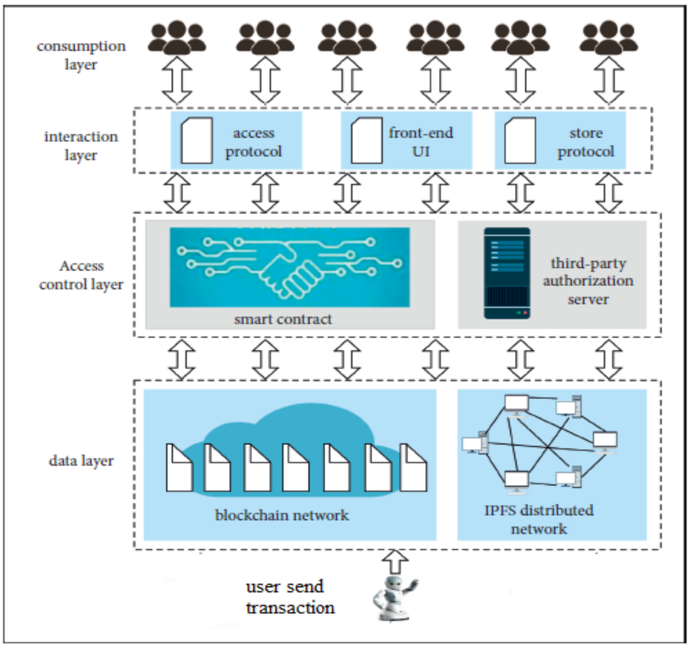

Fig. 3Architectural Diagram of the OHFB Scheme

The consumption layer contains all kinds of data consumers, software, or hardware. The interaction layer provides good access protocols and services for data consumers. The access control layer is the concrete realization of business logic, including the realization of smart contracts and the data control access algorithm. Finally, the data layer consists of the blockchain network and IPFS distributed network. The bottom layer is input user layer. User sends all kinds of data. The blockchain network is used to store the hash values of data, the content hash values generated by IPFS, the access control policy, timestamps, and other information. Data consumers must meet the stipulations of the access control policy on the blockchain to access the data. After successful access, the consistency and integrity of the data can also be verified through the blockchain.

The IPFS distributed network is composed of several server devices with good performance. The data send by user will be encrypted and stored in IPFS, and the hash content values generated by IPFS will be stored on the blockchain. In this way, the blockchain network is associated with IPFS, thereby reducing the burden of blockchain storage. The third-party authorization server mainly generates and transmits the public key (PK) and master key (MK) generated by the CP-ABE initialization algorithm and the private key (SK) generated by the CP-ABE key generation algorithm. In this architecture, users do not need to join the blockchain network, let alone consider the specific implementation details of the whole access control scheme. User will send data or files, and data consumers only need to access data or files. In this scheme, the owner of the data is called the data owner.Block chain is composed of many blocks, each of which contains a block header and block body. The structure of a blockchain is presented in Figure 4.

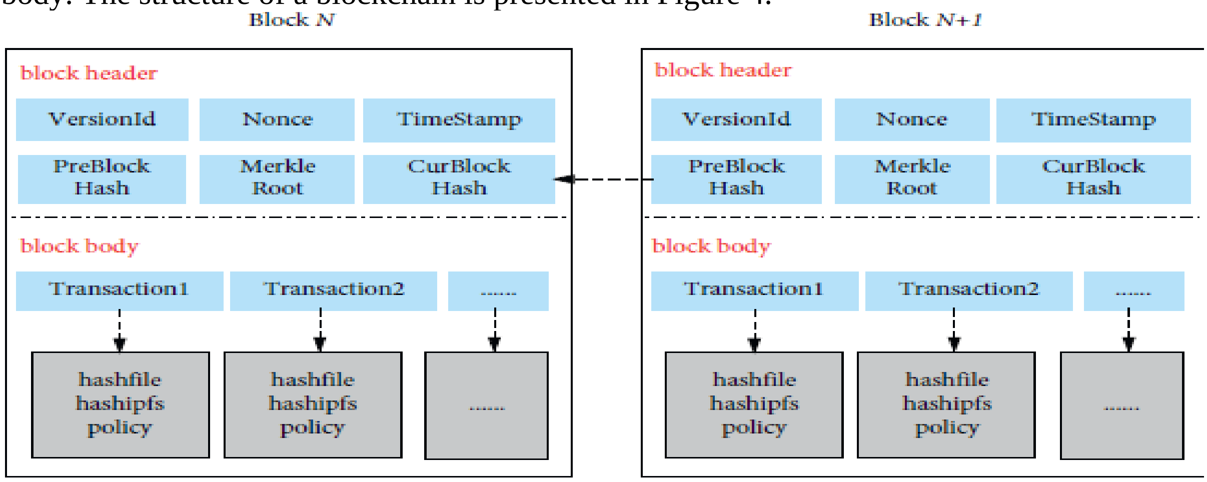

Fig. 4Blockchain Structure

The Merkel root, the version number (VersionId), the timestamp (TimeStamp), and the hash value of the prior block (PreBlock Hash) are all stored in the block header. TimeStamp indicates the block's generation time, PreBlock Hash connects disconnected blocks into chains, and the Merkle root is the hash of several transaction data. When transaction data is manipulated, the Merkle root becomes inconsistent, which can be utilized to verify transaction integrity.The primary hash values of the source data (hashfile), the IPFS hash values (hashipfs), and the access control mechanism (policy) are stored in the block body. These include hashfile, which is 32 bytes in size and is produced by the SHA256 algorithm; hashipfs, which is 32 bytes in size and is the hash value returned after uploading the file to the IPFS network; and policy, which is the data owner's access control strategy; different owners have different policies, with a 1000 bytes upper limit.

Optimized Hyperledger Fabric Transaction Process: A typical Hyperledger Fabric transaction process is discussed for data sharing in the cloud. Before joining the Hyperledger obtain a legal certificate, and then use the certificate to interact with the blockchain network through the command line or Fabric-SDK. Hyperledger supports a multichannel mechanism. Each channel maintains an independent blockchain ledger. Blocks are distributed according to the channel ID, and data between channels is completely isolated. The user needs to specify the channel ID (Channel 1 or Channel 2) when initiating a transaction and send the transaction proposal to the endorsing node. After the endorsement node processes the request, the client returns a signed endorsement response. The client then combines the endorsement responses from different endorsement nodes and sends the final transaction content to the ranking service cluster for processing. After the sorting is completed, the sorting node will distribute it to all master nodes in the channel according to the transaction channel. The master node will synchronize the blocks in the organization. After receiving the transaction, each node verifies the transaction content and signature and adds the legal transaction to the blockchain ledger.

In the hyperledger transaction, suitable placement of transactions into blocks can improve system throughput and reduce delay. An optimization model by crossover operator is introduced in this work for transaction placement to blockchain. Based on time analysis and a transaction flow graph, the fitness is computed to decide which shard is the most suitable one to place a transaction. The transaction flow graph is defined by G = (V, E) , where V and E represent the vertex set and the edge set in the graph, respectively. V consists of transactions. If there are multiple parent transactions, it may be placed in the shard which has the most parent transactions. In addition, the more transactions in the waiting queue, the more time it will take for subsequent arrival transactions. If the time is higher than the transactions are swapped between the blocks via the crossover operator.

Attribute-Based Encryption Mechanism: In the ABEmechanism, the sender uses a set of attributes a to encryptthe message, and the receiver uses a set of attributes a′ todescribe the identity corresponding to the private key. Onlywhen the intersection number of a′ and a exceeds thethreshold value t set by the system can the message receiverdecrypt the ciphertext. However, this mechanism is limitedby access control structures that can only support thethreshold policy [33].

CP-ABE Mechanism: The ciphertext of CP-ABE is associated withaccess control, and its key is associated with the attribute set.Only when the user’s attribute set satisfies this access controlstructure can it be decrypted. Moreover, the access controlauthority of CP-ABE is controlled by the message sender. Therefore, the CP-ABE encryption scheme was adopted in thepresent work to ensure the data owner’s control over the dataand realize the fine-grained access control of the data. It iscomposed of four polynomial algorithms [33].

(1) Initialization Phase: the trusted key distributioncenter executes the random initialization algorithm,as shown in equation (2); the input is the securityparameter r, and the output includes the public key(PK) and master key (MK),

$(PK, MK) = Setup(r)$(2)

(2) Key generation stage: the trusted key distributioncenter executes the key generation algorithm, as shownin equation (2); the inputs are PK and MK generatedby equation (3) and the user-defined attribute set A,and the output includes a private key (SK),

$SK = KeyGen(PK, MK, A)$ (3)

(3) Data encryption stage: the data owner executesthe encryption algorithm, as shown in equation (4);the inputs include PK, the message m to beencrypted, and the access structure T, and the outputis the ciphertext c,

$c = Encrypt(PK, m, A)$ (4)

(4) Data decryption stage: the data requester executesthe decryption algorithm, as shown in equation(5); the inputs include PK, SK, and c, and the output is the plaintext message m,

$m = Decrypt(PK, c, SK)$ (5)

IPFS combines the distributedhash table (DHT), incentive block exchange, self-authenticationnamespace, and other technologies. Moreover, thedata of IPFS are distributed on different devices, and thereexist multiple backups to avoid a single point of failure.Different from the existing web system, in which resourcesare accessed through URLs, IPFS allows for the retrieval offiles by obtaining a unique hash value from the file content. Therefore, once the content of the file changes, the address ofthe file will change, thereby achieving tamper-proof data.With the passage of time, the storage space required by theblockchain will become increasingly larger. In the proposedmethod, the ciphertext of the file is stored in the IPFSnetwork, which can alleviate the rapid expansion of theblockchain caused by too much data.

Data Storage: The data storageprocedure of this scheme includes five participants, namely,the data owner, system server side, IPFS distributed network,blockchain network, and third-party authorization server. The detailed process is as follows.

The data owner (Owner) selects the file to be stored and sets the access control policy of the file (policy). The data consumer (Consumer) can successfully access the file only if the set of attributes of the data consumer meets the stipulations of the access control policy, policy⟵(Owner, file).\
The data owner has a unique AES key (key). If thedata owner has not generated the key before, theserver side calls the AES key generation algorithm togenerate the key, as shown in equation (6). Then, theserver side calls the AES encryption algorithm toencrypt the file and obtain the encrypted file(encfile), as shown in equation (7). Finally, the serverside calls the IPFS storage algorithm to store theencrypted file in the IPFS distributed network, asshown in equation (8), and records the hash value(hashipfs) used to access the ciphertext,

$key = AES.Gen(owner)$ (6)

$encfile = AES.Enc(key, file)$ (7)

$hashipfs = IPFS.Store(encfile)$ (8)

The server side calls the SHA256 algorithm to hashthe file to get the file hash value (hashfile), as shownin equation (9). Then, the previously generatedhashfile, hashipfs, and policy are sent to the blockchainnetwork,

$hashfile = SHA256.Hash(file)$ (9)

The blockchain network receives the data storagerequest and triggers the storage smart contract(StoreCont) to store the hashfile, hashipfs, and policy on the blockchain,

$StoreCont(hashfile, hashipfs, policy)$ (10)

The server side requests the public key (PK) from thethird-party authorization server for the later encryptionof the file.

The data owner has a unique public key (PK) and aunique master key (MK). If the data owner has notgenerated the public key and master key before, thethird-party authorization server will call the initializationalgorithm (Setup) of the CP-ABE algorithmto generate and store PK and MK, as shown inequation (11), and will then send PK to the serverside,

$PK, MK = CP - ABE.Setup(r)$ (11)

The server side calls the encryption algorithm(Encrypt) of the CP-ABE algorithm, takes the policyand PK as the input of the encryption algorithm,encrypts the key to get the ciphertext of the key(enckey), and stores it, as shown in the following equation (12),

$enckey = CP - ABE.Encrypt(PK, key, policy)$ (12)

Data access process: Data access process of this scheme includes five participants, namely, the data consumer, system server side, IPFS distributed network, blockchain network, and third-party authorization server. The detailed process is as follows.

(1) The data consumer (Consumer) sends out a request to access the file, which contains the attribute set A of the data consumer.

(2) After receiving the request from the data consumer, the server side requests the hash value of the file (hashfile) and the hash value used to access the ciphertext of the file to the IPFS network (hashipfs) from the blockchain network.

(3) The blockchain network receives the data access request, triggers a query smart contract (QueryCont), gets the hashfile and hashipfs, and sends them to the server side:

hashfile, hashipfs⟵QueryCont(file)(13)

(4) The server side requests the public key PK and private key SK from the third-party authorizationserver to later decrypt the file.

(5) According to PK, MK, and attribute set *A* of the data consumer, the third-party authorization server executes the key generation algorithm (KeyGen) of the CP-ABE algorithm to generate the private key (SK), as shown in equation (14), and sends PK and SK to the server side,

SK = CP-ABE.KeyGen(PK,MK,A)(14)

(6) According to the hashipfs obtained from the chain, the server side calls the IPFS query algorithm to obtain the ciphertext of the file (encfile) from the IPFS network, as shown in the following equation,

encfile = IPFS.Query(file, hashipfs)(15)

(7) The server side obtains the AES key ciphertext (enckey) that encrypts the file locally and calls

thedecryption algorithm (Decrypt) of the CP-ABE algorithm to decrypt enckey and obtain the decryptionkey (deckey), as shown in the following equation,
deckey = CP-ABE.Decrypt(PK, enckey, SK), enckey = Get(file).(16)
(8) According to deckey, the server side calls the AES decryption algorithm to decrypt encfile and obtains the decrypted file (decfile), as shown in the following equation,
decfile = AES.Dec(deckey, encfile). (17)
(9) The server side calls the SHA256 algorithm to hash decfile and gets the hash value of decfile (dechash), as shown in equation (18). If hashfile and dechash are the same, the access is successful:
dechash = SHA256.Hash(decfile).(18)

4. COMPARISON AND ANALYSIS

In this section, implement the proposed scheme on the Hyperledger Fabric platform and evaluate its performance. Especially, simulate different numbers of medical institutions to construct the blockchain platform for testing the performance and scalability of OHFB-PBAA system.Compared OHFB-PBAA system with the PBAA scheme, conditional identity-based broadcast PRE (CIBPRE) scheme [34] does not support multi-recipient data sharing and the delegation mechanism is not very flexible as only one condition can be specified in the delegation key. Compared with the PBAA scheme, however, the fine-grained delegation is not achieved in CIBPRE, that is, data owners are not able to formulate access policies to specify which ciphertexts needed to be shared. In [21], the proposed an identity-based broadcast PRE(IBPRE) scheme that allows a data owner to share encrypted data with a new group of users and the ciphertexts in the scheme can be re-encrypted for multiple times.

In the Hyperledger Fabric framework, a smart contract is called a Chaincode. It runs in an independent and secure Docker container and initializes and manages the ledger state via the transaction submitted by the application. A smart contract works automatically. Once the smart contract is verified, the verified result set is sent to the Orderer nodes, and the changes in the running results will be shared or synchronized to all Peer nodes in the Fabric network. Hyperledger Fabric provides four basic commands to manage the life cycle of smart contracts, namely, package, install, instantiate, and upgrade. In this experiment, there were two main smart contracts, namely, storage and query smart contracts. (The storage smart contract primarily stores the file hash result encrypted by SHA256 (hashfile), the file hash value generated by IPFS (hashipfs), and the access control policy on the blockchain. The query smart contract mainly extracts metadata, such as hashfile and hashipfs.

For testing purposes, leverage the docker to build a consortium blockchain platform with Hyperledger Fabric, which is constructed using Cloudsimulator in the Cloud environment. The scheme was implemented in a client/server model: the client was built on a Windows 10 PC with a 3.0 GHz Intel Core i7-9700 CPU and 16 GB RAM, and the server was built on a 64-bit dualcore 16GB-memory Windows server in Alibaba Cloud ECS (https://www.alibabacloud.com/product/ecs).In the implementation, the idea of key encapsulation is followed for backwards compatibility. Thus, used 128-bit AES keys to encrypt real data and then encrypted the AES keys with the PBAA's encryption algorithm. The data set used in experiments is chosen from the medical images of Edinburgh Dermofit Library \ (https://licensing.eri.ed.ac.uk/i/software/dermofitimage library.html). Besides, since the PBAA scheme allows sharing data with multiple recipients by specifying multiple conditions, experiments were conducted with varying number of recipients and varying number of conditions.

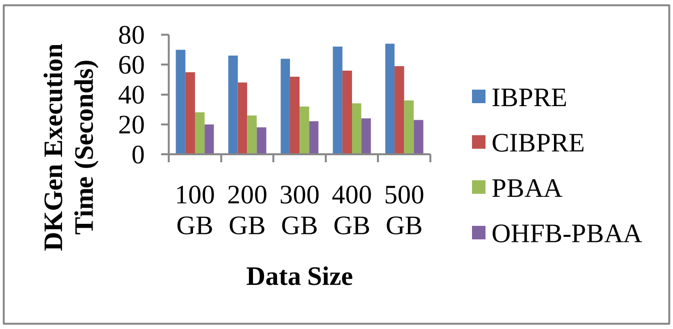

Fig. 5 Execution Time of DKGen for 200 Receivers

Fig. 5 shows the time cost by the delegation algorithm of PBAA, OHFB-PBAA, CIBPRE, and IBPRE. Since a data owner can specify an access policy for a delegation key, conducted this experiment with an access policy of the ^{l} form which is set as 100. From Figure 5, it is demonstrate that the encryption time is linear to the size of data to be encrypted; and for the same data size, the encryption to 2000 receivers consumed (about 20 seconds) more time than the encryption to 1000 receivers. The proposed OHFB-PBAA schema has lesser DKGen execution time of 23 seconds for 500 GB, the other methods such as IBPRE, CIBPRE, and PBAA has takes more execution time of 74 seconds, 51 seconds, and 59 seconds respectively (Refer Table 1).

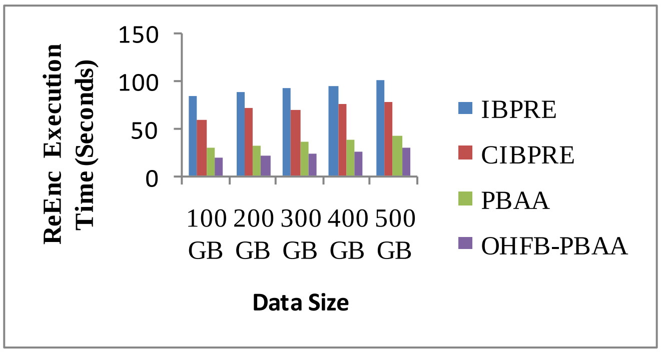

Fig. 6 Execution Time of ReEnc for 200 Receivers

Fig. 6, it seen that the re-encryption timeis independent of the size of data, but grows as the number ofreceivers increases for PBAA, and OHFB-PBAA has needs only 42 seconds, and 30 seconds respectively. But it becomes dependent for CIBPRE, and IBPRE algorithms, so it needs higher time of 102 seconds, and 78 seconds for 500 GB. It can be seen that the re-encryption time is independent ofthe data size since the cloud does not need to process the encrypteddata but only re-encrypts the ciphertext of the encapsulation key(i.e., AES key). As the number of recipients increases, the timecost by the re-encryption algorithm grows(Refer Table 1)

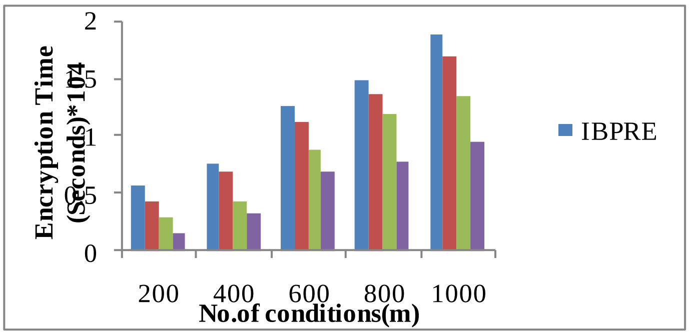

Fig. 7Encryption Time of Schemas for No. of Conditions

Figure 7, Figure 8, Figure 9 and Figure 10 show the execution times of the encryption, delegation, re-encryption and decryption algorithms of the PBAA, OHFB-PBAA, CIBPRE, and IBPRE schemes. Suppose that there are 1000 receivers specified to decrypt a ciphertext in both the schemes and 1GB data is chosen for encryption. Results conducted this comparative experiment with different number of conditions, varying from 100 to 1000 with interval 100. From Figures 7-10, it can be seen that the times cost by the CIBPRE, and IBPRE scheme are dramatically higher than those of OHFB-PBAA scheme. This is because that the parameter m (the number of conditions) is a multiplicative factor of the time consumed by each algorithm of CIBPRE, and IBPRE , but an additive factor of the time consumed by the encryption, delegation and re-encryption algorithms of OHFB-PBAA, and PBAA (the time cost by the decryption is independent of m) (Refer Table 1). From the figure 7, it concludes that the proposed OHFB-PBAA scheme has lesser encryption time of 9515 seconds, whereas other methods such as OHFB-PBAA, CIBPRE, and IBPRE scheme has higher encryption time of 18725 seconds, 16912 seconds, 13461 seconds for 1000 receivers(Refer Table 1).

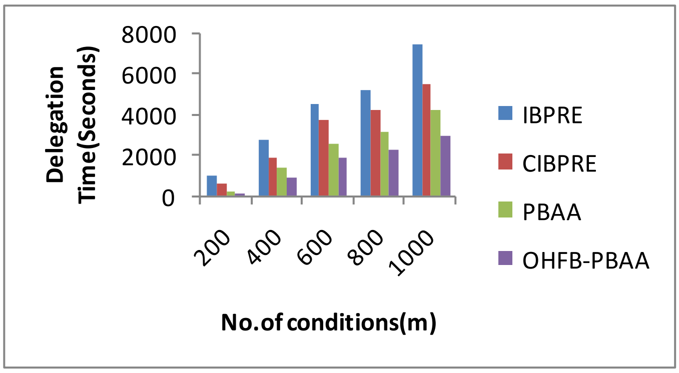

Fig. 8 Delegation Time of Schemas for No. of Conditions

From the figure 8, it concludes that the proposed OHFB-PBAAscheme has lesser delegationtime of 2983 seconds, whereas other methods such as OHFB-PBAA, CIBPRE, and IBPRE scheme has higher delegationtime of 7528 seconds, 5482 seconds, 4258 seconds for 1000 receivers(Refer Table 1).

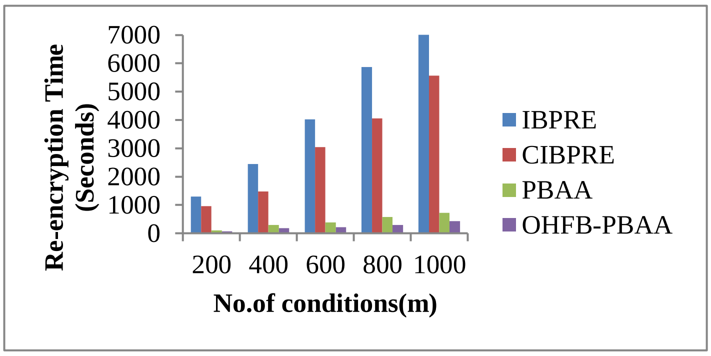

Fig. 9 Delegation Time of Schemas for No. of Conditions

From the figure 9, proposed OHFB-PBAAscheme has lesser re-encryptiontime of 432 seconds, whereas other methods such as OHFB-PBAA, CIBPRE, and IBPRE scheme has higher delegationtime of 7824 seconds, 5571 seconds, 727 seconds for 1000 receivers(Refer Table 1).

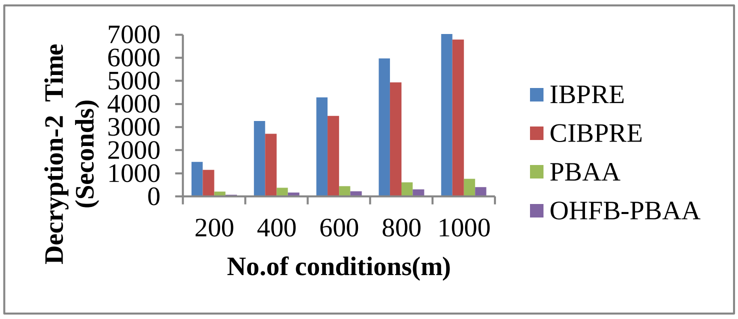

Fig. 10 Delegation Time of Schemas for No. of Conditions

From the figure 10, it is demonstrate that the proposed OHFB-PBAA scheme has reduced decryption-2 time of 406 seconds , whereas other methods such as OHFB-PBAA, CIBPRE, and IBPRE scheme has higher delegation time of 7563 seconds, 6789 seconds, 758 seconds for 1000 receivers(Refer Table 1).

Table 1. Time Comparison of Security Methods

Data Size	**Execution time(seconds)**			
	IBPRE	**CIBPRE**	**PBAA**	**OHFB-PBAA**
100 GB	70	55	28	20
200 GB	66	48	26	18
300 GB	64	52	32	22
400 GB	72	56	34	24
500 GB	74	59	36	23
Data Size	**ReEnc Execution time(seconds)**			
	IBPRE	**CIBPRE**	**PBAA**	**OHFB-PBAA**
100 GB	85	60	29	18
200 GB	88	72	32	22
300 GB	92	70	36	24
400 GB	96	75	38	26
500 GB	102	78	42	30
No.of conditions(m)	**Encryption Time(seconds*10^4)**			
	IBPRE	**CIBPRE**	**PBAA**	**OHFB-PBAA**
200	0.5612	0.4251	0.2917	0.1531
400	0.7528	0.6815	0.4315	0.3215
600	1.2636	1.1258	0.8736	0.6912
800	1.4782	1.3621	1.1834	0.7823
1000	1.8725	1.6912	1.3461	0.9515
No.of conditions(m)	**Delegation Time(seconds)**			
	IBPRE	**CIBPRE**	**PBAA**	**OHFB-PBAA**
200	1025	587	212	132
400	2758	1875	1365	878
600	4528	3782	2615	1853
800	5251	4258	3158	2254
1000	7528	5482	4258	2983
No.of conditions(m)	**Re-Encryption Time(seconds)**			
	IBPRE	**CIBPRE**	**PBAA**	**OHFB-PBAA**
200	1297	963	105	72
400	2451	1474	298	181
600	4025	3042	382	218
800	5871	4061	572	297
1000	7824	5571	727	432
No.of conditions(m)	**Decryption-2 Time(seconds)**			
	IBPRE	**CIBPRE**	**PBAA**	**OHFB-PBAA**
200	1492	1151	215	68
400	3269	2718	378	167
600	4285	3489	452	219
800	5978	4935	612	305
1000	7563	6789	758	427

5. CONCLUSION

In this paper, focus on the secure and flexible data sharing in cloud computing. For the first time, a novel algorithm called OHFB-PBAA approach is introduced to data sharing on cloud computing. OHFB-PBAA approachis proposed, which combines blockchain,CP-ABE, and PBAA. The data owner encrypts hissharing data and stores it on IPFS to maximizedecentralization, and OHFB allows the data ownersto have fine-grained access control over their data.Moreover, it supports revoking permissions of a specificdata user at an attribute level without affectingothers. Before outsourcing data to the cloud,the data owner can specify a group of users and encrypt datawith the identities of these users. The proposed OHFB-PBAA schema, aim at providinga much more flexible data sharing scheme in which a data ownercan generate a re-encryption key by formulating an access policyover multiple conditions.Decryption key of the shared data will be encrypted with CP-ABE according to the specific access policy,and the data owner uses blockchain to publish his data-related information and distribute keys for data users. Only the datauser whose attributes meet the access policy can download and decrypt the data.The proposed scheme isprovably secure and theoretical and experimental analyses revealthe efficiency and practicality of the scheme.Some researchers have proposed using blockchain to solvethe fairness problem in searchable encryption algorithm, it will be considered as scope of future work.

REFERENCES

[1] Y. Wang, Y. Ding, Q. Wu, Y. Wei, B. Qin, and H. Wang, “Privacy preserving cloud-based road condition monitoring with source authentication in VANETS”, *IEEE Transactions on Information Forensics and Security*, Vol. 14, No. 7, pp. 1779-1790, 2019.

[2] H. Yin. Z. Qin, J. Zhang, L. Ou, F. Li, and K. Li, “Secure conjunctive multi-keyword ranked search over encrypted cloud data for multiple data owners”, *Future Generation Computer Systems,* Vol. 100, pp. 689–700, 2019.

[3] I. S. Farahat, A. S. Tolba, M. Elhoseny and W. Eladrosy, “A secure real-time internet of medical smart things (IOMST)”, *Computers & Electrical Engineering,* Vol. 72, pp. 455-467, 2018.

[4] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang and P. Liljeberg, “Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach”, *Future Generation Computer Systems*, Vol. 78, pp. 641-658, 2018.

[5] Y. Zhang, M. Qiu, C. Tsai, M. M. Hassan and A. Almari, “Health-CPS: healthcare cyber-physical system assisted by cloud and big data”, *IEEE Systems Journal,* Vol. 11, No. 1, pp. 88-95, 2015.

[6] Microsoft HealthVault: Web-based personal health record created by Microsoft to store and maintains health and fitness information. Available at: http://www.healthvault.com. Accessed from 4 October 2007 – 20 November 2019.

[7] Google Health: Branch of Google, designed to create a repository of health records and data (personal health record services) to connect doctors, hospitals and pharmacies directly. Available at: https://www.google.com/health, Accessed from 20 May 2008 – 1 January 2019; Again it was restarted in 2018 as a new division, but was later reorganized back into Google in 2021.

[8] A. Ghazvini, and Z. Shukur, “Security challenges and success factors of electronic healthcare system”, *Procedia Technology,* Vol. 11, pp. 212–219, 2013.

[9] Z. Guan, Z. Lv, X. Du, L. Wu and M. Guizani, “Achieving data utility-privacy tradeoff in Internet of medical things: A machine learning approach”, *Future Generation Computer Systems,* Vol. 98, pp. 60-68, 2019.

[10] M. Elhoseny and A. Abdelaziz, “A hybrid model of internet of things and cloud computing to manage big data in health services applications”, *Future Generation Computer Systems,* Vol. 86, pp. 1383–1394, 2018.

[11] Q. Liu, G. Wang, and J. Wu, “Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. Information sciences”, Information sciences, Vol. 258, pp.355-370, 2014.

[12] X. A. Wang, F. Xhafa, J. Ma, and Z. Zheng, “Controlled secure social cloud data sharing based on a novel identity based proxy re-encryption plus scheme, *Journal of Parallel and Distributed*

Computing, Vol. 130, pp.153-165, 2019.

[13] J. Shao, G. Wei, Y. Ling, and M. Xie, "Identity-based conditional proxy re-encryption,"*IEEE International Conference on Communications (ICC),* pp. 1–5, 2011.

[14] H. Deng, J. Zhang, Z. Qin, Q. Wu, H. Yin and A. Castiglione, " Policy-based Broadcast Access Authorization for Flexible Data Sharing in Clouds", *IEEE Transactions on Dependable and Secure Computing,* pp.1-13, 2011.

[15] Y. Fan, S. Liu, G. Tan, and F. Qiao, "Fine-grained access control based on trusted execution environment," *Future Generation Computer Systems,* pp. 551–561, 2018.

[16] Y. Fan, S. Liu, X. Lei, K. C. Li, and G. Tan, "One enhanced secure access scheme for outsourced data," *Information Sciences,* pp. 230–242, 2020.

[17] M. Jemel and A. Serhrouchni, "Decentralized access control mechanism with temporal dimension based on blockchain," *in 2017 IEEE 14th International Conference on e-business Engineering (ICEBE),* pp. 177-182, 2017.

[18] J. Lai, Y. Mu, F. Guo, W. Susilo, and R. Chen, "Anonymous identity based broadcast encryption with revocation for file sharing," *in Australasian Conference on Information Security and Privacy,* Springer, pp. 223–239, 2016.

[19] J. Lai, Y. Mu, F. Guo, W. Susilo, and R. Chen, "Fully privacy-preserving and revocable ID-based broadcast encryption for data access control in smart city", *Personal and Ubiquitous computing,* Vol. 21, No. 5, pp.855-868, 2017.

[20] L. Zhang, H. Xiong, Q. Huang, J. Li, K. K. R. Choo and J. Li, "Cryptographic solutions for cloud storage: Challenges and research opportunities", *IEEE Transactions on Services Computing,* Vol. 15, No. 1, pp. 567 – 587, 2019.

[21] C. Ge, Z. Liu, J. Xia, and L. Fang, "Revocable identity-based broadcast proxy re-encryption for data sharing in clouds", *IEEE Transactions on Dependable and Secure Computing,* Vol. 18, No. 3, pp.1214-1226 2019.

[22] H. Deng Z, Qin, Q. Wu, Z. Guan, and Y. Zhou, "Flexible attribute-based proxy re-encryption for efficient data sharing", *Information Sciences,* Vol. 511, pp. 94–113, 2020.

[23] H. Deng Z. Qin, Q. Wu, Z. Guan, R. H. Deng, Y. Wang, and Y. Zhou, "Identity-based encryption transformation for flexible sharing of encrypted data in public cloud", *IEEE Transactions on Information Forensics and Security,* Vol. 15, pp. 3168–3180, 2020.

[24] S. Yin H. Li, and L. Teng, "A novel proxy re-encryption scheme based on identity property and stateless broadcast encryption under cloud environment", *International Journal of Network Security,* Vol. 21, No. 5, pp. 797–803, 2019.

[25] Q. Huang, Y. Yang and J. Fu, "Precise: Identity-based private data sharing with conditional proxy re-encryption in online social networks", *Future Generation Computer Systems,* Vol. 86, pp. 1523–1533, 2018.

[26] C. Ge, L. Zhou, J. Xia, P. Szalachows and C. Su, "A secure fine-grained identity-based proxy broadcast re-encryption scheme for micro-video subscribing system in clouds", *in International Symposium on Security and Privacy in Social Networks and Big Data,* Springer, pp. 139–151, 2019.

[27] J. Kim W. Susilo, M. H. Au, and J. Seberry, "Adaptively secure identity-based broadcast encryption with a constant-sized ciphertext", *IEEE Transactions on Information Forensics and Security,* Vol. 10, No. 3, pp. 679–693, 2015.

[28] X. Qin, Y. Huang, Z. Yang, and X. Li, "A Blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing", *Journal of Systems Architecture,* Vol. 112, pp.101854, 2021.

[29] H. Gao, Z. Ma, S. Luo, Y. Xu, and Z. Wu, "BSSPD: a blockchain-based security sharing scheme for personal data with fine-grained access control", *Wireless Communications and Mobile Computing,* Vol. 2021, No. 6658920, pp.1-2, 2021.

[30] N. Eltayieb, R. Elhabob, A. Hassan, and F. Li, "A blockchain-based attribute-based signcryption scheme to secure data sharing in the cloud", *Journal of Systems Architecture,* Vol. 102, pp.1-13 2020.

[31] A. Alniamy and B.D. Taylor, "Attribute-based access control of data sharing based on

hyperledger blockchain," *in Proceedings of the 2020 the 2nd International Conference on Blockchain Technology*, pp. 135-139, 2020.

[32] N. Deb, M. A. Elashiri, T. Veeramakali, A. W. Rahmani, and S. Degadwala, "A Metaheuristic Approach for Encrypting Blockchain Data Attributes Using Ciphertext Policy Technique", *Mathematical Problems in Engineering*, Vol. 2022, No. 757996, pp. 1-10, 2022.

[33] X. Lu, S. Fu, C. Jiang, and P. Lio, "A fine-grained IoT data access control scheme combining attribute-based encryption and blockchain", *Security and Communication Networks,* Vol. 2021, No. 6658920, pp. 1-20, 2021.

[34] P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin, "Conditional identity-based broadcast proxy re-encryption and its application to cloud email", *IEEE Transactions on Computers*, Vol. 65, No. 1, pp. 66–79, 2015.