The Journal of Supercomputing (2022) 78:7700–7728
https://doi.org/10.1007/s11227-021-04179-4

Blockchain-based cloud storage system with CP-ABE-based access control and revocation process

Pratima Sharma[1] · Rajni Jindal[1] · Malaya Dutta Borah[2]

Accepted: 26 October 2021 / Published online: 3 January 2022

Abstract

Cloud system provides an on-demand and low-cost computing and storing model. Many organizations and individual end-users are using cloud storage services to back up their crucial data. However, this storage utility suffers from various threats and security issues. Before outsourcing the data to the cloud server, some data security measures should be imposed to ensure security. The blockchain is an advanced technology that stores data in a distributed manner and provides a more secure environment. Therefore, we propose a blockchain-based framework with the Ciphertext Policy Attribute-based Encryption algorithm to provide access control and user revocation methods in the cloud storage system to resolve the above issues. Our scheme offers three main features to provide a secure environment. First, a java-based blockchain network is designed to register data owners and attribute authority using a key generation algorithm. Second, the data owners and attribute authorities store the public information in the blockchain structure, set access policies, and generate the user's secret key to resolve key escrow problems. Third, the immediate attribute modification is deployed to attain fine-grained access control with the user revocation process. The experimental results, analysis, and performance evaluation show that our scheme provides a feasible and reliable environment.

Keywords Blockchain · Cloud storage · CP-ABE · Access control · Revocation

✉ Pratima Sharma
sharmapratima9818@gmail.com

[1] Department of Computer Science and Engineering, Delhi Technological University, Delhi, India

[2] Department of Computer Science and Engineering, National Institute of Technology, Silchar, Assam, India

1 Introduction

Nowadays, data are the main asset. Many electronic devices such as mobile phones, computers, cameras, and laptops generate a massive amount of data each day which needs more storage space and resources. Therefore, the cloud storage system is required to manage such a massive increase in the data and fulfill the storage requirements. The cloud storage system has distributed data centers or servers that utilize virtualization technology to work together and provide storage resources. Recently, the cloud storage system attains massive attention from business organizations and individual users because it is convenient and efficient. Many users and organizations outsource their data on the cloud storage system to alleviate the burden of storage and maintenance in the local storage [1]. The cloud storage system provides on-demand and flexible storage services to an individual or business using a third party over the Internet. The cloud service providers are responsible for maintaining cloud servers and providing services to the users to store and process their data. The cloud storage system allows users to access the data anywhere, thereby supporting an on-demand and pay-per-use model. The user can rent and pay the storage and computation services based on the requirement with the help of the cloud storage model. The cloud provides many benefits and functionality to cloud users, such as low-cost storage, flexibility, automatic update, disaster tolerance, etc. [2–4]. However, it is important to protect user privacy [5] and ensure data protection [6] as data may leak while users store their data in the cloud. Also, users lose control over the data after outsourcing to the cloud, and cloud data may not be safe and vulnerable to various attacks [7–10]. Furthermore, the existing distributed cloud storage system stores the data in multiple data centers or servers in a distributed manner, but they are not completely distributed. Several data centers store the data at high density. Thus, a large amount of data will be exposed if one of the servers or data centers is compromised [11]. Public media worldwide have repeatedly documented unauthorized access concerns related to cloud storage, such as users' private files leaked on iCloud [12]. Unfortunately, there are no viable solutions for the security of cloud systems yet [12]. However, current cloud storage architectures have other flaws, such as centralized data storage, which compromises server security, and the need for trusted third parties affects user privacy [13].

Encryption algorithms provide security features such as confidentiality and data access control. However, attaining access control poses a significant challenge. Bethencourt was the first to implement the CP-ABE algorithm in 2007. According to the CP-ABE algorithm, the ciphertext is linked to an access structure, and the user private keys are derived from attributes [14]. However, the access control mechanism in the existing cloud system requires one or more completely trusted attributes or central authorities to maintain the access policy. If the central authority is compromised, the whole structure is affected. Also, the user revocation where re signature generation is required remains a major task. As a result, decentralized systems are critical in access control and user revocation to eliminate trusted center authority's possible threat. Therefore, to improve the performance of existing applications, there is a need to integrate cloud storage and blockchain technology to propose a blockchain-based distributed cloud storage architecture that can provide secure and reliable cloud storage services for enterprises or individual users. Blockchain is a decentralized database defined as a linked chain of blocks and is difficult to tamper with, forge, or trace [15, 16]. The blockchain stores all transaction information, and almost no one can alter the data once it has been entered. This immutable feature is obtained from the blockchain system and the process itself, not from a specific operation. As a result, blockchain technology is simpler to use and more stable compared to other security technologies. For example, in [17], the authors explain how to use blockchain technology in Intrusion Detection Systems (IDSs), and [18] uses blockchain to protect user data. We use blockchain technology in cloud storage architecture to provide a more secure environment to the users. Instead of concentrating resources in a single data center or server, a blockchain network distributes them among nodes [19]. Although some people have researched blockchain-based security schemes in recent years, most suggest a mechanism or concept for such systems. There is no straightforward approach for realizing the convergence of blockchain technology's decentralization concept with security methods. This is an environment where there is still a lot of work to be done. As a result, blockchain-based decentralized cloud storage system with security methods research is valuable and essential.

To deal with the disadvantages and challenges mentioned above, we designed Java-based blockchain architecture with access control and user revocation process for the cloud storage system to provide a privacy-preserving environment. The proposed architecture implements CP-ABE to provide a key generation mechanism using bilinear mapping-based cryptography and perform data access control mechanisms. Data owners and attribute authorities manage the key-related and user access policy details in a distributed manner by utilizing the blockchain structure and provide a more robust environment. The designed architecture stores publicly accessible information in the blockchain network while also performing access control of stored cloud data. The blockchain network keeps Meta details of users and tracks all access and validation records. By using blockchain technology, we achieve decentralization with security without a trusted central authority.

The paper contributions are as follows:

1. A decentralized and secure blockchain-based architecture is proposed for the cloud storage systems, combining a java-based blockchain network with a cloud storage system and ensuring security features with a revocation process without involving any trusted authority.
2. The proposed architecture addresses the key escrow problem by using two authorities for the key generation process. Here, attribute authority and data owner are responsible for generating user secret keys using their master keys. It provides a distributed approach to generate key-related information, user access policy, and revocation process details without any single authority.

3. The designed java-based blockchain network deploys the immediate user revocation process rather than periodically. It uses the re-encryption approach using the CP-ABE algorithm to update attribute group keys. It also resolves the backward/forward secrecy by deploying an immediate attribute level revocation process. The proposed architecture achieves fine-grained access control at the system level. It allows other attribute group users to access the data even if they are revoked until they satisfy access policies.
4. The performance evaluation, experimental results, and security analysis regarding key escrow and user revocation of the proposed architecture show the system's capability.

The remainder of the paper is set out as follows. Related work is summarized in Sect. 2. Section 3 covers the fundamentals of blockchain technology and access structure. The architecture's system model and implementation details are described in Sect. 4. Section 5 evaluates the architecture performance. Finally, the results, as well as future research directions, are presented.

2 Related work

This section presents the review work relating to traditional and blockchain-based access control and revocation techniques in the cloud storage system.

2.1 Traditional cryptographic techniques

Many researchers have proposed schemes to improve security, quality of services parameters, key management, and many more [20–22]. Traditional cryptographic solutions are not feasible to provide secure access control and revocation process, so a new cryptographic approach, Attribute-Based Encryption (ABE) algorithm, was developed by Sahai and water [23]. In such a method, the user who satisfies the access policy has the right to obtain the plaintext. The attribute-based encryption algorithm is presented in two categories. First, Key-Policy Attribute-based Encryption (KP-ABE) method utilizes the user attributes to describe the data during the encryption process, and user keys are generated from access policies. Second, the Ciphertext Policy Attribute-based Encryption (CP-ABE) approach defines the user registration credentials using an attribute list in which access policies are used during the decryption process. Since then, various variations of ABE algorithms have been proposed [24–31]. For example, in [24], the authors propose a hybrid ABE algorithm that allows direct and indirect revocation processes. Direct revocation enables the user to specify the revocation process during the encryption process, whereas indirect revocation is implemented using the key update process. The proposed architecture takes advantage of both methods. It lacks the details of the collusion attack and key escrow problem. Attrapadung et al. [25] present a novel KP-ABE technique to achieve fine-grained access control with constant ciphertext length. The introduced architecture first uses identity-based broadcast encryption to output monotonic access structure using generic transformation. Then, the non-monotonic access structure is designed using the previous step output without affecting efficiency. The novel scheme uses the hybrid method to provide a non-monotonic access structure, increasing the revocation process's complexity. Several research studies have evaluated the methods and identified the taxonomy of distributed certificate authorities used for the revocation process [26, 27]. Masdari [28] presents a reliable and secure revocation certificate in a mobile adhoc network. The presented method is based on the false accusation problem of the local certificate revocation method by verifying neighboring nodes and issuing accusations. Datta et al. [29] used the ABE algorithm for Boolean circuits to design a direct revocation approach. The suggested architecture supports decryption policies to achieve the revocation process and reduces the process complexity from linear to logarithm using multilinear maps. In [30], the authors suggest an unrestricted revocation process by using a subset difference mechanism. The recommended method deploys an ABE approach with the subset method to decrease the computation cost and increase efficiency. The method only considers the monotonic access structure of the attribute-based scheme.

Furthermore, Liu et al. [31] use a KP-ABE algorithm to support black box traceability and revocation. The proposed method is based on a monotonic access structure for defining access policies and supports a large attribute set. It lacks the technique to prevent key escrow problems and does not support non-monotonic access structures. Nieto et al. [32] utilize the revocation predicate encryption algorithm, which specifies the ciphertext's access policies in the form of decryption policies. It ensures the attribute hiding property derived from the ABE scheme to achieve privacy features. In [33], the authors design a direct revocation model with two KP-ABE algorithms. It suggests a method to revoke the user's one attribute rather than whole attributes without affecting the private key. The user can still decrypt the data until the unrevoked attributes meet the condition. The deployed architecture designs two access trees for the same user, first for the non-revoked user decryption process and the second for the revocation list. The concept of two access trees unnecessary increases the computational overhead. Jia et al. [34] present an identity-based signature method to outsource the cloud server's revocation process. The revocation process key update process is transferred to the revocation cloud server, which handles all related activities. The suggested scheme uses a time update key scenario rather than an immediate revocation approach. In [35, 36], authors suggest data access control methods in the cloud computing system using an improved attributed-based technique to enhance the data security. The authors present the various access control models and highlight the advantages and disadvantages. The suggested scheme utilizes the searchable encryption method to share the resources in a cloud environment where one-upload and many-download service is required. The scheme lacks the revocation procedure and is dependent on a trusted third party to generate and manage user keys. However, existing ABE techniques suffer from key escrow problems and utilize the centralized authority to maintain user-related details.

2.2 Blockchain-based techniques

Blockchain is a relatively advanced computer technology development where members can immediately capture and share transactions with other members [37]. Several research studies have used the software to correct the flaws of the current attribute-based access process. For example, in [38], the authors present a blockchain-based data sharing scheme using smart contract and ABE to achieve user revocation process. The proposed architecture provides privilege management during the data sharing process using attribute level revocation. It involves trusted authority for the key management process and encrypts or decrypt data; thus, the user data's security is at risk. In case of failure of key management center, users cannot access their information, and the entire structure will get affected. Su et al. [39] present a blockchain-based healthcare system to protect users' privacy by using the attribute-based signature method for the user revocation process. The architecture deploys attribute master key and update-key concepts to link with the user's identity and attribute set to generate the signing key. It uses the KUNodes algorithm to achieve attribute revocation in the healthcare systems without involving any central authority. The designed architecture client nodes are overloaded as all the key-related operations are done at the user nodes. Moreover, in [40], the authors propose a new ABE technique using blockchain technology to outsource the decryption process securely. The proposed architecture uses a smart contract to ensure the proxy entity's reward for the successful outsources decryption process. It also utilizes the sampling technique to allow the miners to check the validity of the decryption result. However, the suggested scheme uses the ABE technique to guarantee only the secure outsource decryption process rather than the revocation method.

Yu et al. [41] propose a blockchain-based selective revocation method using dynamic accumulators and signature concepts. The suggested approach uses anonymous authentication for smart industrial applications to ensure attribute privacy and selective revocation. It deploys on the Ethereum platform to support multi-authority security and allows users to denote multiple attributes. The suggested method is specifically designed for industrial applications. In [42], the authors propose a blockchain-based application in higher education to issue academic certificates. It also incorporates a revocation process to revoke diploma certificates that have been issued incorrectly. The model uses blockchain structure to store revocation data which increases overhead because of block size limitation. In [43], the authors propose a blockchain-based architecture for a multi-server system to provide privacy-preserving authentication mechanisms and efficient revocation processes. It allows users to have a smart card to access multiple servers, and miners play the role of permission server's use in the registration process. The suggested architecture uses a re-registration scheme to support user revocation in case of smart card is lost. The architecture mainly focuses on a single registration center with forward security. Most studies use the key management center to generate a secret key for the user using the master key and attributes. However, it provides an advantage of removing the requirement of public key infrastructure to store and manage keys. Still, it is a challenging task because the key management center involves a key escrow problem as it generates secret keys for users without any users' involvement. Also, the key management center can act maliciously and hamper user security.

Ning et al. [44] develop a cryptcloud framework to secure cloud storage systems. The designed architecture uses the CP-ABE algorithm to support the white box traceability, provide auditability, and revocation process. Similarly, in [45], the authors design a multi-authority-based CP-ABE approach to ensure the revocation process in the Named Data Network (NDN). The suggested method deploys the proxy-based access control technique with forward and backward security. Also, Fan et al. [46] deploy the access control mechanism using proxy servers for smart cities. It also deploys the CP-ABE scheme to achieve the security features and user revocation process. Wang et al. [47] design a secure cloud storage system using blockchain technology. The designed framework deploys the Ethereum platform to set a valid access period for cloud users and achieves an access control mechanism. It uses the smart contract functionality to store ciphertext in the blockchain network. Similarly, Saini et al. [48] propose a smart contract-based access control framework using blockchain technology for the healthcare system. The designed system uses the Elliptic Curve Cryptography to encrypt the healthcare data before storing it on the cloud. However, these techniques provide different methods to achieve access control and user revocation process. Still, they suffer from many disadvantages, such as the involvement of a semi-trusted party which affects the overall security of the architectures. Also, it requires more processing overhead due to the involvement of proxy servers. Table 1 summarizes the existing techniques.

3 Preliminaries

In this section, background studies of the proposed scheme have been discussed in detail.

3.1 Overview of blockchain

Blockchain is mostly considered the core technology of Bitcoin cryptocurrency, developed by an unknown person Nakamoto in 2008 [49]. In essence, blockchain technology is a peer-to-peer network that servers as a public trusted and shared ledger. This innovative technology has recently emerged as a popular technique for academicians and researchers that it has the potential to develop blockchain-based applications beyond Bitcoin cryptocurrency [50, 51]. The key focus of blockchain technology is decentralization that indicates that the blockchain is shared throughout the network nodes. Each network node has the authority to check the operation of other nodes in the network and generate, verify, and validate the new transactions of the blockchain network. The blockchain decentralization architecture provides reliable and secure operations with tamper resistance and no single point failure features. Generally, the blockchain is categorized into two categories, public and private blockchain networks. The public blockchain is accessible to everyone, which means anyone can join and generate transactions and participate in the consensus mechanism (e.g., Bitcoin network). However, the private blockchain is a permissioned network where all participants have to take permission from the authority to participate or create transactions in the blockchain network [52]. Figure 1 depicts the blockchain structure in which each block contains

Table 1 Strength and weakness of related work

References	Findings	Technique used	Features	Limitations
[38]	Design a blockchain-based data sharing model to support attribute revocation	CP-ABE	Attribute-based access control with attribute level revocations	Involve trusted third party as the key management center
[39]	Propose a blockchain-based healthcare system with an attributed-based signature method for the revocation process	Attribute-based Signature Scheme	Protect user identity with the attribute revocation process	Overloads the user node due to key-related processing overhead
[40]	Suggest a fair outsourced ABE decryption method based on blockchain	Pairing-based Attribute Encryption	Ensure fairness between the user and the proxy with secure decryption	Only focus on the decryption process
[41]	Present blockchain-based anonymous authentication with revocation method	Zero-Knowledge Proof of Knowledge protocol (ZKPoK)	Anonymous authentication with selective revocation process	Specifically designed for the authentication process in smart industrial applications
[42]	Deploy revocation process for academic certificates stored on the blockchain	Blockchain with InterPlanetary File System (IPFS)	Issue academic certificates securely and also revoke the invalid issued digital certificates	Increases the blockchain structure overhead due to storage of certificates
[43]	Combine blockchain with multi-server architecture to provide authentication and revocation features	Ouroborros algorithm	Provide mutual authentication, forward security, user anonymity, revocation, etc	Multiple security requirements increase the complexity of the architecture
[44]	Propose a secure access control method for the cloud storage system	CP-ABE	Auditing, secure access control with malicious cloud users revocation features	The architecture is dependent on semi-trusted authority
[45]	Design a muti-authority-based access control scheme for Named Data Network (NDN)	CP-ABE	Proxy-assisted access control feature with attribute revocation process	The involvement of proxy servers increases processing time
[46]	Suggest a proxy-assisted access control mechanism of cloud data for smart cities	CP-ABE	Efficient access control with decryption process and achieves forward and backward security	Dependent on trusted certificate authority for the management of key-related information

Table 1 (continued)

References	Findings	Technique used	Features	Limitations
[47]	Present decentralized cloud storage using blockchain technology	CP-ABE	Distributed access control	Depends on the semi-trusted cloud storage system and does not include non-negated details in the access tree
[48]	Suggest access control method for IoT-based healthcare system using smart contracts	ECC	Provide decentralized access control	Lacks revocation procedure and suffers from scalability issues
Proposed Work	Propose a fully distributed java-based customized blockchain network to achieve access and revocation in the cloud storage system without involving any trusted authority	CP-ABE using bilinear mapping	Achieve fine-grained access control, immediate user revocation process with backward and forward security, provide attribute update feature, and resolve key escrow problem	The proposed scheme lacks the integrity checking feature

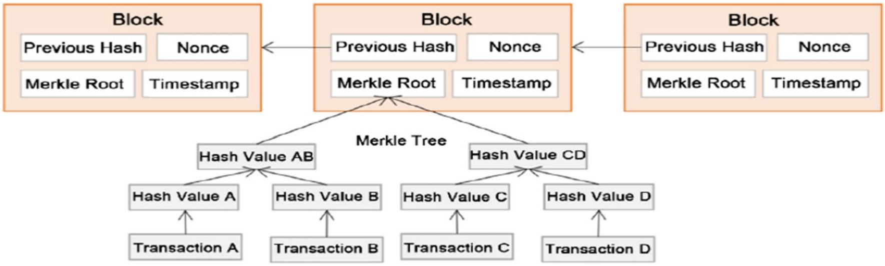

Fig. 1 Blockchain structure

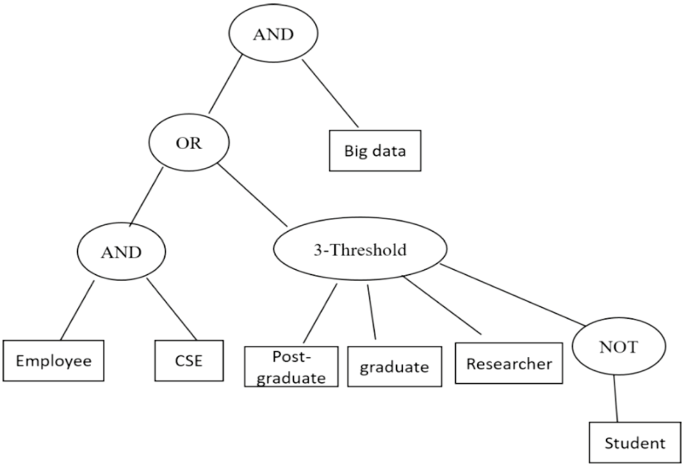

Fig. 2 Access tree

the previous block's hash, and Nonce denotes the solution to the proof of work puzzle. The timestamp represents the block generation time, and the Merkle root authenticates all the transactions.

3.2 Access tree

The access policy is represented in the form of tree structure A_t. The access tree non-leaf nodes are designed using the Threshold gate. The proposed architecture uses AND, OR, and K-Threshold gates. The access tree leaf nodes denote attributes for both negated and non-negated user details. The existing ciphertext attribute-based encryption algorithm allows to use NOT gates only at the tree structure leaf nodes. Thus, the proposed architecture allows a user to provide non-monotonic access control. Figure 2 represents an illustration of an employee's access structure who needs to give access permission to other employees. The leaf nodes denote the non-negated attributes employee, CSE, staff, big data, researcher, guest faculty, and one negated attribute student. The first attribute set is {employee, CSE, big data}. It

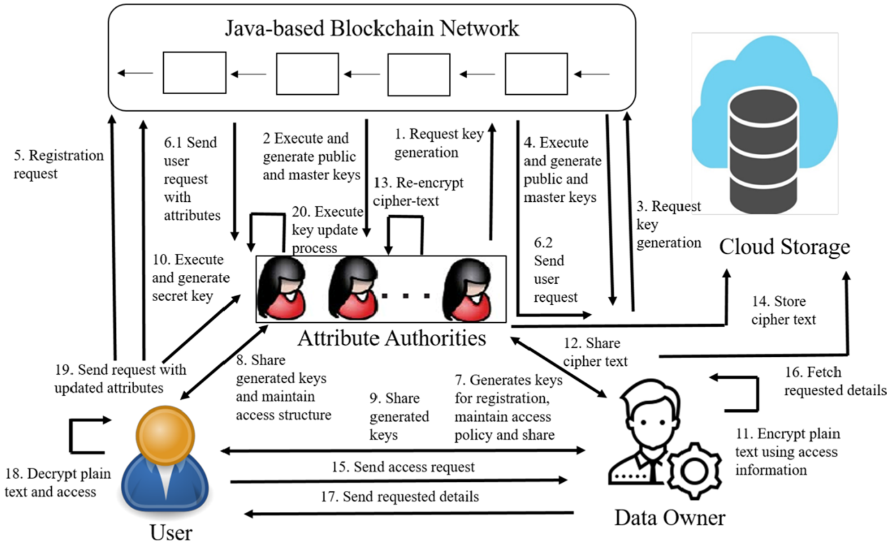
Java-based Blockchain Network
Cloud Storage
5. Registration request
6.1 Send user request with attributes
2 Execute and generate public and master keys
1. Request key generation
13. Re-encrypt cipher-text
20. Execute key update process
4. Execute and generate public and master keys
3. Request key generation
6.2 Send user request
10. Execute and generate secret key
Attribute Authorities
14. Store cipher text
12. Share cipher text
19. Send request with updated attributes
8. Share generated keys and maintain access structure
9. Share generated keys
7. Generates keys for registration, maintain access policy and share
16. Fetch requested details
11. Encrypt plain text using access information
18. Decrypt plain text and access
15. Send access request
17. Send requested details
User
Data Owner

Fig. 3 System model

represents the computer science branch employee having research area big data is an authorized access policy set. Whereas the second attribute set {student, post-graduate, graduate, researcher} represents unauthorized access policy set means graduate, post-graduate, and researcher students are not allowed to access employee's data.

4 Proposed architecture

In this section, the entire proposed scheme has been discussed in detail.

4.1 System model

Figure 3 shows the decentralized privacy-preserving architecture for blockchain-based cloud storage. The proposed system model consists of five core entities:

1. *Blockchain*: Blockchain maintains the transparent, tamper-proof structure to keep the general details of the users. Therefore, the data transfer between the users of the proposed architecture is non-tamper and transparent. The blockchain network ensures access control and revocation functionality using various smart contract functions such as key generation, encryption, re-encryption, decryption, and key update functions.
2. *Attribute authority:* Attribute authority produces the user keys using the CP-ABE algorithm. Attribute authority has the right to generate, distribute, and modify user attribute keys. Attribute authority also manages user access rights based on

his attributes. Attribute authority employs a re-encryption process to attain fine-grained access control to apply the attribute level's user revocation method.

3. *Data owner:* The data owner outsources the data on the cloud storage to effectively manage data distribution. The data owner also defines the attribute-based access policies and uses these policies to encrypt the user files.
4. *User:* Users can access the data when their attributes satisfy the ciphertext's access policies. The user can use the decryption algorithm to decrypt the file using the secret key and obtain the plain text. If the data owner removes any user, then the user cannot access that group data.
5. *Cloud storage:* Cloud storage stores encrypted files uploaded by the data owners. It manages the access of stored data and gives relevant services.

The proposed architecture workflow, as shown in Fig. 3 explanation, is as follows:

1. Attribute authorities and data owners send key generation requests to generate global parameters and public and master keys to register in the proposed architecture.
2. The blockchain network executes the key generation function and generates a public key and master key for the data owner and attribute authority.
3. The user sends a registration request to the blockchain network, which sends the user details to the data owner and attribute authority with attribute list like userID, department, email id, contact number, address, DOB, etc.
4. The data owner and attribute authority save user details and generate an access policy corresponding to the user's attribute list using the CP-ABE algorithm. The generated keys of the data owner and attribute authority are shared with the user. The user executes the key generation algorithm to generate a secret key using data owner and attribute authority generated keys.
5. The data owner encrypts the plain text using the access structure to generate the ciphertext. The generated ciphertext shares with the attribute authority for the re-encryption process to provide user revocation and generates header information so that revoked users cannot access the encrypted ciphertext.
6. After the re-encryption process, the ciphertext with header information is outsourced to the cloud server.
7. The user sends an access request to the data owner. In response, the data owner fetches the requested details from the cloud server and shares them with the user.
8. Then, the user uses the decryption process to decrypt the plain text. If the user is authenticated, he can decrypt the data as the ciphertext associated with the header information that only allows active users to access it.
9. The user may send the request to attribute authority to add, remove or modify the attributes.
10. The attribute authority executes the key update function to regenerate the keys for the modified user list and updates the group access policy.

The proposed architecture consists of the following smart contract algorithms:

Table 2 Notation table

Notations	Description
GP	Global parameters
PK_{do}	Data owner public key
MK_{do}	Data owner master key
PK_{aa}	Attribute authority public key
PK^{*}_{aa}	Attribute group public key
MK_{aa}	Attribute authority master key
SK_{do,U_t}	Data owner secret key for the user
SK_{aa,U_t}	Attribute authority secret key for the user
U_t	User ID
SK_{U_t}	User secret key
SK^{*}_{aa,U_t}	User secret key for attribute group
CT	Ciphertext
CT'	Re-encrypted ciphertext
$Header$	Ciphertext header information for the user access
K	Security parameter denotes group size
F	File
T	Attribute set
U_t	User ID
A_t	Access structure
A_G	Attribute group
DO	Data owner
AA	Attribute authority

$Setup\left(,\mathbb{G},g,e,\mathbb{G}_{T,}H,H_1\right)\rightarrow GP$: This algorithm computes the public parameter GP in the setup phase.

$DOkeygenerate(GP)\rightarrow\left(PK_{do},MK_{do}\right)$: This algorithm generates public key PK_{do} and master key MK_{do} for data owners.

$AAkeygenerate(GP)\rightarrow\left(PK_{aa},MK_{aa}\right)$: This algorithm generates public key PK_{aa} and master key MK_{aa} for attribute authorities.

$DOkeycomp\left(MK_{do},U_t\right)\text{and}AAkeycomp\left(MK_{aa},U_t,S_i,S\right)\rightarrow SK_{do,U_t}\text{and}SK_{aa,U_t}$: These two algorithms are executed by the user U_t. First, data owners take master key MK_{do} and user-id U_t as input and output unique private key SK_{do,U_t} as output for the user. Then, attribute authorities take a master key MK_{aa}, user-id U_t, confidential data T_i and attributes set T described as input and outputs SK_{aa,U_t} user key. Finally, the user gets the final secret key SK_{U_t} by combining these two key generation algorithms.

$Encrypt(F,PK_{do},PK_{aa},A_t)\rightarrow CT$: This algorithm takes file F, public keys PK_{do}, PK_{aa}, and access structure A_t, as inputs and outputs ciphertext CT.

$ReEncrypt\left(CT,A_G,PK^{*}_{aa}\right)\rightarrow CT'$: This algorithm takes the CP-ABE generated ciphertext CT, attribute group details A_G, attribute group public key PK^{*}_{aa} and outputs ciphertext CT'. The only active user who satisfies the updated policy can access the updated ciphertext.

$Decrypt\left(CT', SK_{U_t}, K\right) \rightarrow F$: This algorithm is executed by the active users to decrypt the ciphertext and obtains plain text F. All the notations and their descriptions used in the proposed scheme are given in Table 2.

4.2 Smart contract functions

The proposed architecture deploys the various smart contract functions using a CP-ABE algorithm [23] to provide various services to the user such as key generation process, data outsource service, access control, and revocation process using re-encryption process. The proposed architecture involves data owner and attribute authority entities to provide the essential services to the user using bilinear mapping. The data owner selects a bilinear group $\mathbb{G}$ of prime order p and generator g and two random numbers $a, b \in \mathbb{Z}_p$. Let p be a prime number and $\mathbb{G}, \mathbb{G}_T$ be multiplicative cyclic groups of order p. A security parameter K denotes the group's size. We also use Lagrange coefficients $\Delta_{i,\mathcal{L}}$ for any $i \in \mathbb{Z}_p^*$ and a set, $\mathcal{L}$, of elements in $\mathbb{Z}_p^*$: define $\Delta_{i,\mathcal{L}}(x) = \prod_{j \in \mathcal{L} j \neq j} \frac{x-j}{i-j}$. The hash functions are also designed $H : \{0,1\}^* \rightarrow \mathbb{G}$ to link each attribute with random group element in $\mathbb{G}$ and $H_1 : \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$, to model random group element. A map e: $\mathbb{G} \times \mathbb{G} \rightarrow \mathbb{G}_T$ satisfying the following properties is called a bilinear map or bilinear pairing.

- $e\left(u^a, v^b\right) = e(u,v)^{ab}, \forall u, v \in \mathbb{G},\ \ a, b \in \mathbb{Z}_p$
- If g is a generator of $\mathbb{G}$, then $e(g,g)$ is a generator of $\mathbb{G}_T$
- $e(u,v)$ is efficiently computable for all $u, v \in \mathbb{G}$
- Let $H : \{0,1\}^* \rightarrow \mathbb{G}$ be a hash function that maps attribute to the random element of $\mathbb{G}$.

We denote this bilinear map by $(p, \mathbb{G}, g, e, \mathbb{G}_{T,} H, H_1, K)$, where g is a generator of $\mathbb{G}$.

4.2.1 Key generation function

This phase generates global parameters GP at the initial stage. It uses a bilinear group $\mathbb{G}$ of prime order p and generator g based on security parameters. The two hash functions $H : \{0,1\}^* \rightarrow \mathbb{G}$ and $H_1 : \mathbb{G}_T \rightarrow \mathbb{Z}_p^*$ are also selected from the hash functions, universal family. Algorithm 1 is executed by the data owners. Line 1 selects $\dashv$ random number from the finite field over prime p, $\mathbb{Z}_p^*$ and use a generator to generate public and master key for the data owner. It outputs two keys pubic key PK_{do} and master key MK_{do}. Both keys are saved in the blockchain network using the SHA256 hashing algorithm.

```
Data Owner Key Generation Algorithm
Algorithm 1: $DOkeygen(GP\ )$
Input: Global Parameter $GP$, Random Exponent $a$
Output: Public and Private keys $PK_{do}, MK_{do}$
    1. Begin
    2. Let $a \in \mathbb{Z}_p^*$      //selects a random number
    3. $\mathfrak{b} \leftarrow g^a$          // g is a generator
    4. $PK_{do} \leftarrow \mathfrak{b}$
    5. $MK_{do} \leftarrow a$
    6. End
```

Algorithm 2 is executed by the attribute authorities. Line 2 selects the random number β from $\mathbb{Z}_P^*$. Similarly, Line 3 selects the random number ⌋ from $\mathbb{Z}_P^*$. After the random number selection, the public key is calculated by applying a mapping function using the selected random number β. The master key is generated using the generator function using β, and the attribute group public key is created using ⌋. It generates public keys PK_{aa}, and PK_{aa}^*, and master key MK_{aa} for the attribute authority, and generated details are saved in the blockchain structure in the form of the hash using the SHA-256 algorithm.

```
Attribute Authority Key Generation Algorithm
Algorithm 2:$AAkeygen(GP\ )$
Input: Global Parameter $GP$, Random Exponent $\beta$
Output: Public and Private keys $PK_{aa}, MK_{aa}$
    1. Begin
    2. Let $\beta \in \mathbb{Z}_p^*$      //selects a random number
    3. Let $c \in \mathbb{Z}_p^*$      //selects a random number
    4. $PK_{aa} \leftarrow e(g, g)^{\beta}$
    5. $MK_{aa} \leftarrow g^{\beta}$
    6. $PK_{aa}^* \leftarrow g^c$
    7. End
```

4.2.2 User key generation function

The attribute authority and the data owner participate in the user key generation process. First, the attribute authority and data owner verify the user-id and check if it already exists. After verifying the user id, the attribute authority and data owner follow the user key generation process using the secure two Party Computation function (2PC) that allows the two parties to jointly compute the function using their inputs without sharing their inputs with the other party [53]. Thus, the proposed architecture utilizes the 2PC protocol to securely generate the user secret key parameters with the help of the attribute authority and the data owner. Then, the user uses the generated parameters to create the secret key. The attribute authority and data owner execute the key generation algorithm to generate one and two keys,PK_{do,U_t},

SK_{aa,U_t}, andSK^*_{aa,U_t}. The user utilizes these keys to generate his own key SK_{U_t} using Eq. 1. The user uses the generated key to decrypt the ciphertext.

Algorithm 3 generates the secret key for the user using the secret keys generated by the data owners and attribute authorities. First, the data owner and attribute authority authenticate the user details then uses the 2PC function to calculate the value of x using the data owner's N_t and $\dashv$ parameters and attribute authority's M_t and β parameters. Next, the data owner calculates the value of S using the generator function and shares it with the attribute authority using the 2PC protocol. Similarly, the attribute authority utilizes the S to calculate the P parameter. Based on calculated parameters, attribute authority generates the secret key SK_{aa,U_t} and generates the group key SK^*_{aa,U_t} using PK^*_{aa} key parameter c by applying a hash function on user-id U_t. The data owner uses the attribute set T to calculate the value of D using a generator with a random number and generates the secret key SK_{do,U_t}.

```
User Key Generation Algorithm
Algorithm 3:$AAkeygen(\,) \leftrightarrow DOkeygen(\,)$
Input: User id $U_t$
Output: User Secret Key $SK_{U_t}$
Begin
if $AA$ and $DO$ authenticate $User\ U_t, then$
    $DO$ selects random exponent $N_t \in_R \mathbb{Z}_p^*$        //unique and secret to the user
    $x \leftarrow (\beta + N_t)a \leftarrow 2PC\big(DO(N_t, a), AA(\beta)\big)$    //general and secure 2PC computation function
    $AA$ selects random exponent $M_t \in_R \mathbb{Z}_p^*$
    $DO \leftarrow S = g^{\frac{x}{M_t}} = g^{\frac{(\beta+N_t)a}{M_t}}$        //data owner computes and sent to attribute authority using 2PC
    $AA \leftarrow P = S^{\frac{1}{a^2}} = g^{\frac{(\beta+N_t)}{aM_t}}$        //attribute authority computes and sent to data owner using 2PC
    $SK_{aa,U_t} \leftarrow P^{M_t} \leftarrow g^{\frac{(\beta+N_t)}{a}}$        //attribute authority generates secret key
    $SK_{do,U_t} \leftarrow (\forall i \in T{:}\, D_i = g^{M_t}.H(i)^{r_i}, D_i = g^{r_i}), r_i \in_R \mathbb{Z}_p^*$    //data owner generates secret key
    $SK^*_{aa,U_t} \leftarrow H(U_t)^c$        //attribute authority generates another secret key for attribute group key
else
    exit
End
```

The algorithms generate the secret key for the user by using data owner and attribute authority generated secret keys $SK_{U_t} = \{SK_{aa,U_t}, SK_{do,U_t}\}$ as given below:

$$SK_{U_t} = \left(g^{\frac{(\beta+N_t)}{a}}, \left(\forall i \in T : D_i = g^{M_t} \cdot H(i)^{r_i}, D_i = g^{r_i}\right)\right) \tag{1}$$

The user's secret key is denoted by SK_{U_t}. The user uses the generated key to execute the decryption algorithms to decrypt the ciphertext and obtains the plain text that the data owner outsources.

4.2.3 Encryption function

The data owner is responsible for outsourcing the file on the cloud server. To store the file on the cloud server, the data owner needs the access information A_t with universal attributes U and performs encryption to convert plaintext to ciphertext using attribute authority public key PK_{aa}. Algorithm 4 selects a polynomial V_n for each node of the access tree A_t using a top-down approach. Then, the algorithm sets the degree d_n for V_x node that is one less than the threshold value of that node i.e., $d_n = K_n - 1$. For the root node R, the random value is selected from $\mathbb{Z}_p^*$ and set $V_R \leftarrow \int$. Next, for any other node of the tree, the algorithm sets V_x by assigning index values. In the end, the algorithm generates the cipher text using L denotes leaf nodes of access tree A_t, attribute authority public key $e(g,g)^{\beta}$, data owner public key $\mathfrak{b}$, and polynomial function as represented in Line 11.

```
Encryption Algorithm
Algorithm 4:$Encrypt(F, PK_{do}, PK_{aa}, A_t)$
Input: File $F$, Public Key $PK_{do}$, Public Key $PK_{aa}$, and Access Structure $A_t$
Output: Cipher Text $CT$
1.  Begin
2.  $For$ each Node $n\ in\ A_t$
3.      Select a polynomial $V_n$        //start from the root node
4.      $d_n \leftarrow K_n - 1$                //$K_n = V_x$ node threshold value in $A_t$
5.      $if\ x = Root_{node}\ R$
6.          Let $\mathscr{s},\ \mathscr{s} \in_R \mathbb{Z}_p^*$
7.          $V_R \leftarrow \mathscr{s}$
8.     $V_x \leftarrow V_{p(x)}\big(index(x)\big)$        //set node $x$ index value to $V_x$
9.  $end\ for$
10. Let $L \leftarrow Leaf\ Nodes(A_t)$
11. $CT = (A_t, C' = Fe(g,g)^{\beta\mathscr{s}}, C_1 = \mathfrak{b}^{\mathscr{s}}\ \forall\ l \in L{:}\ C_l = g^{V_y^{(0)}}, C_l = H(\delta_l)^{V_y^{(0)}})$
12. End
```

4.2.4 Re-Encryption Function

The re-encryption phase prevents the access of plain text from the revoked users. This function is executed by the attribute authority to re-encrypt the ciphertext by using attribute group A_G before outsourcing to the cloud server—this algorithm control user access according to attributes. Algorithm 4 selects the random number K_G from Z_P^* for each group member G_l present in attribute group A_G and calculates the re-encrypted cipher text CT' as represented in Line 4. Then, the algorithm calculates the header information using randomly selected numbers P^* and R from Z_p^* and attribute group public key PK_{aa}^*. For each group, member belongs to A_G the algorithm calculates the exponent function P. In the end, the header information is calculated as presented in Line 15. The authorized user, when requesting the cloud server, then responds with CT' and Header. Using this information, the user can decrypt the data until the user is not on the revocation list.

Re-Encryption Algorithm

Algorithm 5: $Reencrypt(CT, A_G, PK_{aa}^*)$

Input: Cipher Text CT, Attribute Group A_G, and Attribute Group Public key PK_{aa}^*
Output: Cipher Text CT'

```
Begin
For each $G_l$ in $A_G$
        Let $K_G$, $K_G \in \mathbb{Z}_p^*$
        $CT' = (A_t, C' = Fe(g,g)^{\beta s}, C_1 = h^s \, \forall\, l \in L: C_l = g^{V_y^{(0)}}, C_l = \left(H(\delta_l)^{V_y^{(0)}}\right)^{K_G})$
end for
Let $P^*, R \in \mathbb{Z}_p^*$
$x_t = H_1\left(e\left(Q_t^{P^*}, PK_{aa}^*\right)\right) \forall\, U_t \in G$
For each $G_l$ in $A_G$
        For each user $U_i, i \le n$      //n is the number of users in $G_l$
            $i = i * (x - x_i)$
            $K = K + a_i x^i$
            $P_i \leftarrow g^{a0}$                  // exponent function
      end for
      $f(x) = I = K \bmod p$
     $Header_l = \{K_G.P_0^R, P_1^R...., P_m^R\}$
    $Header = Header_l$
end for
End
```

4.2.5 Decryption function

This function works in two stages, as explained below:

i. Group key decryption stage: The user requests the data owner for the data access. In response, the data owner sends a request to the cloud server and provides cipher text $\left(CT', Header\right)$ to the user. Then, user generates the key using the attributes T associated with the user from *theHeader*. For example, a user U_t having attributes λ_j means $U_t \in G_j$, and the user can obtain the attribute key K_l from *Header* as shown below.

1. Calculates, $x_t = H_1\left(e\left(g^P, PK_{U_t}^*\right)\right)$
2. Calculates $K_l.P_0^R.\prod_{i=1}^{n}\left(P_i^R\right)^{x_t^i} = K_l.g^{Rf^j(x_t)} = K_l$ where n is the number of users in the attribute group G_j.

Then, user updates the secret key by using the generated attribute group K_l using Eq. 2.

$$SK_{U_t} = \left(SK_{aa,U_t}, SK_{do,U_t}\right) = \left(g^{\frac{(\beta+N_t)}{a}}\left(\forall i \in T : D_i = g^{M_t} \cdot H(i)^{r_i}, D_i = (g^{r_i})\right)^{\frac{1}{K_l}}\right) \tag{2}$$

The generated key process is secured as no one can secret key K_l other than the user U_t.

ii. Data Decryption stage: With the help of generated key user can decrypt the ciphertext CT'. The decryption process uses a recursive procedure with node x and its children. The recursive function is defined as $Decryptnode\left(CT', SK_{U_t}, x\right)$ where x denotes the leaf node of access tree A_t as given in Eq. 3.

a. Function $Decryptnode\left(CT', SK_{U_t}, x\right)$

$$if\, \lambda x \in T \text{ and } U_t \in G_x, then$$

$$Decryptnode\left(CT', SK_{U_t}, x\right) = \frac{e\left(D_x, C_x\right)}{e\left(D'_x, C'_x\right)} = \frac{e\left(g^{N_t} \cdot H(x)^{rx}, g^{V_x^{(0)}}\right)}{e\left(\left(g^{rj}\right)^{\frac{1}{K_l}}, \left(H(x)^{V_x^{(0)}}\right)^{K_y}\right)} = e(g,g)^{rtV_x^{(0)}} \tag{3}$$

If $U_t \notin G_x$ or $\lambda_x \notin T$, then the function returns null as shown in Eq. (4).

$$Decryptnode\left(CT', SK_{U_t}, x\right) = e(g,g)^{rtV_x^{(0)}} = NULL \tag{4}$$

b. Function $Decryptnode\left(CT', SK_{U_t}, x\right)$, x is not a leaf node in the access tree A_t The function recursively call all child nodes of x.

$$Q_z = Decryptnode\left(CT', SK\mathbb{c}\right)\{\mathbb{c} \in Childnodes(x)\}$$

$$\mathbb{L}_x\, denotes\ leaf\ nodes\ set\ of\ \mathbb{c}$$

$$Q_z = NOT\, NULL\, if\, \mathbb{L}_x \notin \emptyset$$

$$Q_z = NULL\, if\, \mathbb{L}_x \in \emptyset$$

$$Q_x = \prod_{\mathbb{c} \in \mathbb{L}_x} Q_{\mathbb{c}}^{\Delta_{i.p_x(0)}}$$

$$where\ i = \mathbb{c}\ node\ index\ and\ p_x\ is\ \{index\,(\mathbb{c}) \notin \mathbb{L}_x\}$$

$$e(g,g)^{N_t V_x^{(0)}} = P$$

$$Plain\, text = C' * \frac{P}{SK_{U_t}} = \frac{\left[F * Fe(g,g)^{\dashv s^*} e(g,g)^{N_t.S}\right]}{e\left(h^s, g^{\frac{(\beta+N_t)}{\dashv}}\right)} = F \tag{5}$$

4.2.6 Key update function

In case the user changes the attribute list like adding or removing attributes such as an address, email id, contact number, department, etc. The access permissions for that user should be updated to preserve backward and forward secrecy. The attribute authority executes this smart contract function when the user request is received regarding updating a particular attribute group's attributes. After receiving the user request, the attribute authority first sends the updated attribute group list membership to the data owner to update the stored user-related information at the owner's side. Then, it generates new keys for the updated group attributes, and the updating process is completed. The process does not affect the remaining non-related user's keys due to the changed group attributes. This phase works as follow:

1. The attribute authority selects a random number s' and an attribute key K_l. Perform encryption of CT using PK^{*}_{aa} as shown in Eqs. (6) and (7).
$$CT = A, C = Fe(g \cdot g)^{b(s'+s)}, C = h^{s'+s}, C_i = g^{V_i^{(0)}+s'} \tag{6}$$
$$C_i = \left(H(i)^{V_i^{(0)}+s'}\right)^{K_i} \forall l \in L\backslash\{i\} : C_l = g^{V_y^{(0)}+s'}, C_l = \left(H(l)^{V_l^{(0)}+s'}\right)^{K_l} \tag{7}$$
2. The attribute authority uses a new attribute group to create a polynomial function $f(x)$ for including or excluding users. Then, it creates a new header message by calculating a new $Header_i$ using K_i as given in Eq. (8).
$$Header = \left(g^P, Header_i \forall l \in L\backslash\{i\} : Header_l\right) \tag{8}$$

Whenever a user requests the cloud data, the data owner replies with the header information and ciphertext. The user can only decode the ciphertext when the attribute group satisfies. The above algorithms ensure access permission at various levels and also maintain access restrictions for different users.

4.3 Security analysis

The proposed architecture defined the following security goals to describe the access and revocation process. First, the data owner describes the users' list who can access the cloud data. Then, the access policies are defined according to the user attributes. The proposed architecture achieves the following security features:

1. *Data Protection:* Cloud data access is restricted as per the defined access policies even if the user collides with the other user. The access policies give access permission at all levels and achieve a fine-grained access control mechanism. The proposed architecture ensures data protection from unauthorized users as it allows only users to decrypt data if they have enough attributes. If the user is revoked from the group, he cannot access that group's plain text. The proposed architecture achieved this using the immediate attribute revocation process. The ciphertext is re-encrypted using a group-based attribute access policy instead of the whole access policy. Another possibility of attack may be from the cloud server or attribute authorities. There may be chances that they may share the information for their profit. We deployed two key generation methods to make this process independent of a single authority to resolve this issue. If the user requests

the registration process, then the data owner and attribute authority independently generate separate keys and send them to the user. Then, the user generates the secret key using these keys. Hence, the proposed architecture guaranteed confidentiality and data protection.

2. *Collusion Tolerance:* The proposed architecture avoids a collision attack that is the main security requirement in the ABE algorithm. If multiple users coordinate with each other, then they may decrypt the ciphertext by linking the attributes. Therefore to avoid such type of attack, the attribute authority and cloud server cannot coordinate with the revoked user to any extent. Also, the user uses the unique random value to generate the secret key. For the collusion attack, the attacker should recover the $e(g, g)^{\lfloor s}$ to decrypt the ciphertext. The attacker cannot perform the decryption process until he gets the random value of the user.
3. *Backward and Forward Security:* Backward secrecy means if the new user joins the group, he cannot access the cloud server's data before. Forward secrecy deals with restricting access of revoked users for subsequent cloud data that will be outsourced in the future, except if the user satisfies the access policy to the other valid attributes. The proposed architecture achieves backward and forward security by using an immediate user revocation process instead of timely revocation. If the user discards or updates an attribute in a group, the re-encryption process deploys using a new secret key. Then, the generated key is shared with all the related group users.

5 Result analysis

This section presents the details of the proposed architecture experimental setup, implementation details, performs privacy, and performance evaluation.

5.1 Experimental setup

We designed the proposed architecture in the Java programming language. The experiments are conducted on the Windows 10 operating system, with Intel® Core ™ i7CPU, 2.5 GHz, and 8 GB RAM. We used Netbeans 7.0 IDE to implement the proposed architecture with JDK 1.7. The external auxiliary Java Pairing-based Cryptography is used to implement bilinear pairing-based cryptography in the proposed architecture. For simulating the cloud storage environment, the CloudSim-3.0.3 framework is used [54]. CloudSim provides the cloud computing component system and behavioral modeling. Simulation of cloud environment offers useful insights to explore such dynamic, distributed, and scalable environments. The jar folder cloudsim-3.0.3.jar is used for integrating a java-based blockchain network with the simulated cloud environment. There are many more mature blockchain networks available, like Ethereum and Hyperledger. However, these blockchain networks are not directly deployed in the proposed architecture because the block header is more complicated in the traditional blockchain network. In contrast, we define a minimized block header while maintaining the user's public details. Thus, we have designed and implemented our minimal block structure to combine with multi-server cloud storage and ensure an easy, secure and reliable environment.

5.2 Implementation details

The proposed permissioned blockchain network creates multiple classes such as creating blocks, generating hashes using the SHA-256 algorithm, storing blocks, validating blockchain, etc., using the java programming language. Further, the blockchain network is deployed with CP-ABE algorithm and cloud storage services to provide core functionality such as key management, encryption, re-encryption, key update, and decryption using the smart contract concept. Depending on the event occurred by the users, different smart contract functions execution triggers automatically and provide the service to the user. At the backend, the proposed architecture deploys the cloud storage service using the CloudSim tool. CloudSim is an open software that provides cloud computing data center virtualization technology with various virtualized cloud modeling and simulation functions interfaces. The proposed architecture utilizes the org.cloudbus.cloudsim package to simulate the workload, load balancing, and policy-related implementation using different java classes such as DatacenterBroker and CloudletScheduler Vmallocationpolicy, etc. The proposed architecture created the CloudSim environment of 15 data centers, 50 virtual machines, and 100–1000 task (transactions) by implementing the different classes. The proposed computes the transaction-related attribute information for all the ready tasks. Then, the ready transactions send the request to the load balancer of the virtual machine. Finally, the load balancer balances the resources of the virtual machine and allocates the task accordingly.

The proposed architecture is designed so that it can be easily further extendable to include new functionality or deploys in real-world scenarios. The architecture was developed in three parts. The first part implements the graphical user interface to provide essential services to users. The second part deploys the main logic of the architecture using smart contract functions. Different smart contract functions are defined to achieve different services of the proposed work. Each participant of the blockchain network executes the smart contract function in the form of a transaction that follows blockchain procedure to append it as a block in the blockchain structure. Lastly, the back end of the architecture design uses the CloudSim tool to store ciphertext in the cloud storage. Depending on the proposed work composition, configuration, and deployment requirement, the real cloud environment exhibits varying demand–supply patterns and system size. Moreover, the users have heterogeneous and competing quality of services requirements. Thus, the proposed work uses the simulation to evaluate the performance and test the services of the architecture in a repeatable and controllable environment free of cost, identify the performance bottleneck and handle the complexities that arise. Moreover, the proposed architecture design works independently, allowing the update in one part without affecting the main logic of the proposed work. Therefore, the proposed architecture can be easily extended to include the new functionality or utilize the real cloud platforms such as Amazon, Microsoft Azure, etc., by updating the connectivity classes.

Table 3 Comparison of related work with proposed scheme

Scheme	Authority	Expressiveness	Key Escrow	Revocation
[24]	Key	NA	Yes	Time attribute revocation
[25]	Key	Non-monotonic	Yes	Immediate attribute level
[29]	Single	Monotonic	Yes	Time attribute revocation
[30]	Single	Monotonic	Yes	Subset difference revocation
[31]	Multiple	Monotonic	Yes	Immediate attribute level
[32]	Multiple	NA	Yes	Time attribute revocation
[33]	Key	Monotonic	Yes	Immediate attribute level
Proposed scheme	Multiple	Non-monotonic	No	Immediate attribute level

5.3 Privacy evaluation

Table 3 compares related work with the proposed scheme based on parameters like authority type, access policy, key escrow, and revocation method. Compared to existing work, our proposed architecture used multiple authority systems to generate keys for the users. Furthermore, the architecture used a non-monotonic access policy. Negative attributes define access set attributes that make the access structure clearer compared to other methods. It also employed the re-keying method to implement an immediate attribute revocation approach rather than time-based attribute revocation. This provides a more secure environment for cloud data in terms of forward and backward secrecy. Our scheme achieves fine-grained access control by using the re-encryption technique and using two authorities to compute keys for the user. Also, the used key generation process has solved the key escrow problem with two authorities' help.

Table 4 compares the proposed scheme efficiency with the related work. We perform a comparative analysis of computation cost for generating various keys. First, the communication cost for sending and receiving data between the data owner/authority and the cloud server is measured using ciphertext key size. Second, the user's storage cost is measured from private key size. Last, the attribute authorities' public key size of the system is measured for comparison. From the comparison, we can conclude that our scheme is the most efficient CP-ABE with direct revocation. Our scheme space and computation complexity do not depend on N_u; the total number of users in the system, which is supposed to be huge. Also, the proposed architecture private and public key sizes are smaller than the existing approaches. It requires less computational overhead without involving logarithmic operation. Thus, the proposed scheme is more efficient in all aspects.

Table 4 Efficiency comparison of related work with proposed scheme

Scheme	Ciphertext key size	Private key size	Public key size	Algorithm
[24]	$(1 + A + \log R)E_0 + E_1$	$\left((E_A + 1)\log N_u\right)E_0$	$(\log R + A)E_0 + E_1$	KP-ABE
[25]	$3E_0 + E_1$	$\left((R + 1).E_A\right).E_0$	$(R + A + 1)E_0 + E_1$	KP-ABE
[33]	$(2 + 2A)E_0 + E_1$	$4E_AE_0$	$(3 + 2A + R)E_0$	KP-ABE
[29]	$(2 + A)E_0 + E_1$	$\left(1 + E_A\right)E_0$	$(\log N_u + E_A + 3)E_0$	KP-ABE
[30]	$(16A + 64R - 27)E_0 + E_1$	$(5 + 16E_A + 16$ $(\log^2 N_u + \log N_u)E_0)$	$111E_0 + E_1$	KP-ABE
[32]	$4N_uE_0 + E_1$	$8N_uE_0$	$32N_uE_0$	CP-ABE
[31]	$\left(16\sqrt{N_u} + 3A\right)E_0 + E_1$	$\left(2 + 2A + \sqrt{N_u}\right)E_0$	$(5 + 8\sqrt{N_u})E_0 + \sqrt{N_u} + E_1$	CP-ABE
Proposed scheme	$(2A + 1)E_0 + E_1 + E_A$	$(2K + 2)E_0$	$E_0 + E_1$	CP-ABE

$E_0 \rightarrow$ *Element bit size in* $\mathbb{G}$; $E_1 \rightarrow$ *Element bit size in* $\mathbb{G}_T$; $E_A \rightarrow$ *Access bit size of access tree* A_t

$A \rightarrow$ *Attribute count in* A_t; $N_u \rightarrow$ *User count in attribute group* G; $R \rightarrow$ *Number of revoked users*

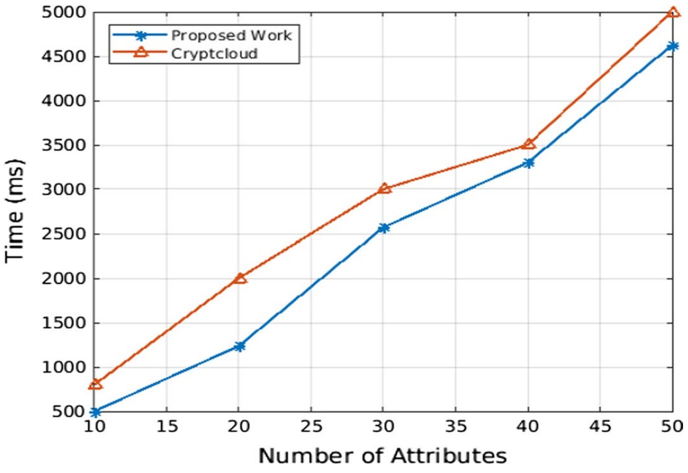

Fig. 4 Key generation time comparison between proposed scheme and cryptcloud

5.4 Performance evaluation

This section analyses the performance of the proposed architecture and also compare it with the cryptcloud [44] scheme, NDN technique [45], and Proxy technique [46]. We consider the encryption, decryption, key generation, and re-encryption functions' performance time for the comparison process. The encryption time denotes the time required to convert the plaintext to ciphertext, whereas decryption time defines the time required to obtain the plaintext from the ciphertext. The re-encryption time involves the time needed to re-encrypt the ciphertext. Furthermore, the key generation time includes the time required to generate the keys for

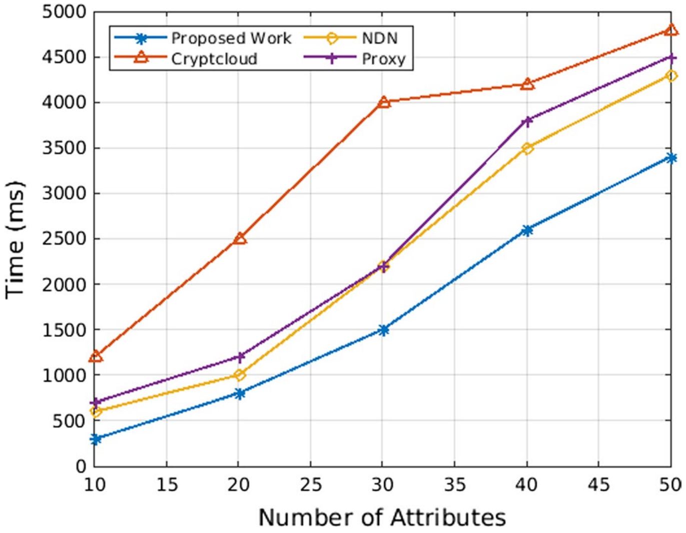

Fig. 5 Encryption time comparison between the proposed scheme and existing techniques

the user. As shown in Fig. 4, we have calculated the key generation time by varying user attributes in the system and compared it with the cryptcloud scheme [44]. We can depict from the analysis that the proposed work requires less time for the key generation process. The proposed scheme used a bilinear-based cryptography approach to generate the secret keys for the user. In contrast, cryptcloud used a semi-trusted key management center to create keys and ciphertext conversion that affected security features. The cryptcloud scheme is a semi-distributed architecture; thus, the proposed scheme provides a better approach.

Figure 5 shows the encryption time comparison between the proposed architecture with cryptcloud [44], NDN [45], and proxy [46] schemes. In all approaches, the encryption time increases with the number of attributes. We can analyze that the proposed system requires less time to perform the encryption process than the existing literature. The proposed scheme uses a robust encryption process with a Pairing-based library (PBC) and a 160-bit elliptic curve group using a supersingular curve in the 512-bit finite field. In contrast, the cryptcloud technique utilizes the symmetric session key to encrypt the plaintext, thus needing more time to share the same key for both encryption and decryption processes. The NDN and proxy schemes involve the proxy servers to perform the encryption and decryption, which increases the overall time for both processes. Similarly, Fig. 6 depicts that the proposed approach's decryption time is less than existing techniques because the decryption process involved in existing work requires more operations to achieve a user-based revocation process.

Figure 7 depicts the time required by the proposed work for the re-encryption process and compares it with the NDN technique [45] by varying the number of attributes. It is observed that the increment in the number of attributes also increases the re-encryption time in both approaches. Also, the proposed work

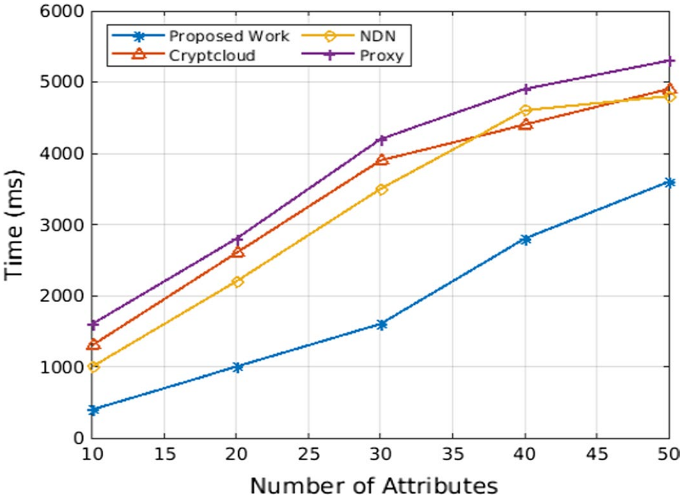

Fig. 6 Decryption time comparison between the proposed scheme and existing techniques

performs better than the existing literature NDN. The NDN technique involves the proxy re-encryption approach that requires the agent module to respond according to the user's request. Therefore, it requires extra pre-processing before the re-encryption process. In contrast, the proposed work directly executes the re-encryption smart contract function for the user request and reduces the extra overhead. Therefore, the performance evaluation and comparative analysis show that the proposed architecture maintains the effectiveness of the existing work and provides a better and secure solution.

6 Conclusion and future work

The privacy and security of outsourced data are key challenging issues in the cloud storage system. The proposed architecture introduced a blockchain-based fine-grained access control method using the CP-ABE algorithm to provide a robust user revocation process in the cloud storage systems. The proposed methodology utilized the two-authority-based key generation scheme to resolve key escrow issues and make the system independent on a single authority. Thus, it is difficult for the attribute authority or cloud servers to misuse the outsourced data. Furthermore, the proposed scheme ensures the outsourced data's privacy and confidentiality by restricting the users from accessing the data without proper credentials. The proposed architecture deployed the immediate attribute level user revocation process rather than time-based to provide scalable access restriction using the CP-ABE algorithm. The performance evaluation, comparative analysis, and experimental results indicate that the proposed architecture offers a more efficient and scalable environment to the outsourced cloud data. For future work, we plan to include the integrity checking process in the proposed architecture, which ensures that the uploaded documents are

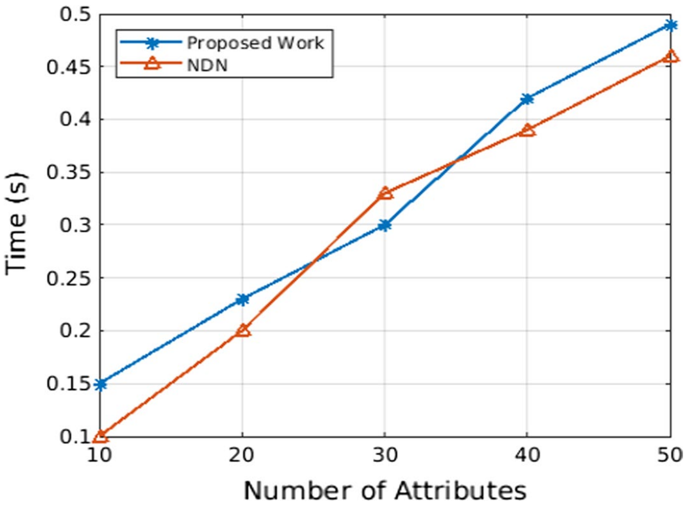

Fig. 7 Re-encryption time comparison between the proposed scheme and NDN technique

not tampered with by malicious users and enhances the security of the architecture. Furthermore, to enhance the liveliness of the architecture, other features such as public verifiable deletion mechanism and distributed payment system can be added to provide a complete distributed cloud storage solution.

References

1. Azhir E, Navimipour NJ, Hosseinzadeh M, Sharifi A, Darwesh A (2019) Query optimization mechanisms in the cloud environments: a systematic study. Int J Commun Syst 32(8):e3940
2. Singh A, Chatterjee K (2017) Cloud security issues and challenges: a survey. J Netw Comput Appl 79:88–115
3. Shin Y, Koo D, Hur J (2017) A survey of secure data deduplication schemes for cloud storage systems. ACM Comput Surv 49(4):1–38
4. Du M, Wang Q, He M, Weng J (2018) Privacy-preserving indexing and query processing for secure dynamic cloud storage. IEEE Trans Inf Forensics Secur 13(9):2320–2332
5. Zhang Y, Chen X, Li J, Wong DS, Li H, You I (2017) Ensuring attribute privacy protection and fast decryption for outsourced data security in mobile cloud computing. Inf Sci 379:42–61
6. Zhang Y, Zheng D, Deng RH (2018) Security and privacy in smart health: Efficient policy-hiding attribute-based access control. IEEE Internet of Things J 5(3):2130–2145. https://doi.org/10.1109/JIOT.2018.2825289
7. Kaaniche N, Laurent M (2017) Data security and privacy preservation in cloud storage environments based on cryptographic mechanisms. Comput Commun 111:120–141
8. Li Y, Gai K, Qiu L, Qiu M, Zhao H (2017) Intelligent cryptography approach for secure distributed big data storage in cloud computing. Inf Sci 387:103–115
9. Shen J, Gui Z, Ji S, Shen J, Tan H, Tang Yi (2018) Cloud aided lightweight certificate less authentication protocol with anonymity for wireless body area networks. J Netw Comput Appl 106:117–123
10. Shen J, Wang C, Li T, Chen X, Huang X, Zhan Z-H (2018) Secure data uploading scheme for a smart home system. Inf Sci 453:186–197. https://doi.org/10.1016/j.ins.2018.04.048
11. Lyengar R (2020) Apple to strengthen security after iCloud nude celebrity photos leak. http://time.com/3271667/apple-jennifer-lawrence-icloud-leak660security/, 2014. Accessed September 4, 2020.

12. Kofahi NA, Al-Rabadi AR (2018) Identifying the top threats in cloud computing and its suggested solutions: a survey. Networks 6(1):1–13
13. Zyskind G, Nathan O, et al (2015) Decentralizing privacy: using blockchain to protect personal data. In: Security and Privacy Workshops (SPW). IEEE, pp 180–184
14. Bethencourt J, Sahai A, Waters B (2008) Ciphertext-policy attribute-based encryption. In: Proceeding of the IEEE Symposium on Security and Privacy (SP), pp 321–334
15. Namasudra S, Deka GC, Johri P, Hosseinpour M, Gandomi AH (2021) The revolution of blockchain: state-of-the-art and research challenges. Arch Comput Methods Eng 28(3):1497–1515
16. Sharma P, Jindal R, Borah MD (2020) Blockchain technology for cloud storage: a systematic literature review. ACM Comput Surv 53(4):1–32
17. Meng W, Tischhauser E, Wang Q, Wang Y, Han J (2018) When intrusion detection meets blockchain technology: a review. IEEE Access 6:10179–10188
18. Lin Q, Yan H, Huang Z, Chen W, Shen J, Tang Yi (2018) An id-based linearly homomorphic signature scheme and its application in blockchain. IEEE Access 6:20632–20640
19. Sharma P, Jindal R, Borah MD (2021) Blockchain-based decentralized architecture for cloud storage system. J Inf Secur Appl 62:1–15
20. Mohajer A, Barari M, Zarrabi H (2018) Big data-based self-optimization networking: a novel approach beyond cognition. Intell Autom Soft Comput 24(2):413–420
21. Masdari M, Ahmadzadeh S, Bidaki M (2017) Key management in wireless body area network: challenges and issues. J Netw Comput Appl 91:36–51
22. Mohajer A, Bavaghar M, Farrokhi H (2020) Mobility-aware load balancing for reliable self-organization networks: multi-agent deep reinforcement learning. Reliab Eng Syst Saf 202:107056
23. Sahai A, Waters B (2005) Fuzzy identity-based encryption. In: Cramer R (eds) Advances in cryptology—EUROCRYPT. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, vol 3494, pp 457–473
24. Attrapadung N, Imai H (2009) Attribute-based encryption supporting direct/indirect revocation modes. In: Parker MG (eds) Cryptography and Coding, IMACC, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, vol 5921, pp 278–300
25. Attrapadung N, Libert B, de Panafieu E (2011) Expressive key-policy attribute-based encryption with constant-size ciphertexts. In: Catalano D, Fazio N, Gennaro R, Nicolosi A (eds) Public Key Cryptography—PKC 2011, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, vol 6571, pp 90–108
26. Masdari M, Jabbehdari S, Ahmadi MR, Hashemi SM, Bagherzadeh J, Khadem-Zadeh A (2011) A survey and taxonomy of distributed certificate authorities in mobile ad hoc networks. EURASIP J Wirel Commun Netw 112:1–12
27. Masdari M, Bidaki M, Naghiloo F (2017) Comprehensive evaluation of the localized certificate revocation in mobile ad hoc network. Wireless Pers Commun 94:977–1001
28. Masdari M (2016) Towards secure localized certificate revocation in mobile ad-hoc networks. IETE Tech Rev 34(5):561–571. https://doi.org/10.1080/02564602.2016.1215270
29. Datta P, Dutta R, Mukhopadhyay S (2015) General circuit realizing compact revocable attribute-based encryption from multilinear maps. In: ISC, vol 9290 of LNCS. Springer, pp 336–354
30. Datta P, Dutta R, Mukhopadhyay S (2016) Adaptively secure unrestricted attribute-based encryption with subset difference revocation in bilinear groups of prime order. In: Pointcheval D, Nitaj A, Rachidi T (eds) Progress in Cryptology—AFRICACRYPT, Lecture Notes in Computer Science. Springer, vol 9646, pp 325–345
31. Liu Z, Wong DS (2016) Practical ciphertext-policy attribute-based encryption: traitor tracing, revocation, and large universe. Comput J 59(7):983–1004. https://doi.org/10.1093/comjnl/bxv101
32. Nieto JMG, Manulis M, Sun D (2012) Fully private revocable predicate encryption. In: Susilo W, Mu Y, Seberry J (eds) Information Security and Privacy, ACISP. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, vol 7372, pp 350–363
33. Wang P, Feng D, Zhang L (2011) Towards attribute revocation in key-policy attribute-based encryption. In: Lin D, Tsudik G, Wang X (eds) Cryptology and Network Security, CANS, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, vol 7092, pp 272–291
34. Jia X, He D, Zeadally S, Li L (2017) Efficient revocable ID-based signature with cloud revocation server. IEEE Access 99:1–9
35. Namasudra S (2021) Data access control in the cloud computing environment for bioinformatics. Int J Appl Res Bioinform 11(1):40–50

36. Namasudra S (2019) An improved attribute-based encryption technique toward the data security in cloud computing. Concurr Comput Pract Exp 31(3):e4364
37. Gai K, Guo J, Zhu L, Yu S (2020) Blockchain meets cloud computing: a survey. IEEE Commun Surv Tutor 22(3):2009–2030. https://doi.org/10.1109/COMST.2020.2989392
38. Ma W, Ma J, Zhang Q, Xue H, Li Y, Dang X, Zhao M, Zhang J, Han C, Wu J (2020) Attribute revocable data sharing scheme based on blockchain and CP-ABE. In: Proceedings of the 4th International Conference on Computer Science and Application Engineering (CSAE 2020), Association for Computing Machinery, New York, NY, USA, pp 1–7
39. Su Q, Zhang R, Xue R, Li P (2020) Revocable attribute-based signature for blockchain-based healthcare system. IEEE Access 8:127884–127896. https://doi.org/10.1109/ACCESS.2020.3007691
40. Zheng H, Shao J, Wei G (2020) Attribute-based encryption with outsourced decryption in blockchain. Peer-to-Peer Netw Appl 13:1643–1655
41. Yu Y, Zhao Y, Li Y, Du X, Wang L, Guizani M (2020) Blockchain-based anonymous authentication with selective revocation for smart industrial applications. IEEE Trans Ind Inf 16(5):3290–3300. https://doi.org/10.1109/TII.2019.2944678
42. Vidal FR, Gouveia F, Soares C (2020) Revocation mechanisms for academic certificates stored on a blockchain. In: Proceedings of the 15th Iberian Conference on Information Systems and Technologies (CISTI), Seville, Spain, pp 1–6. https://doi.org/10.23919/CISTI49556.2020.9141088.
43. Xiong L, Li F, Zeng S, Peng T, Liu Z (2019) A blockchain-based privacy-awareness authentication scheme with efficient revocation for multi-server architectures. IEEE Access 7:125840–125853. https://doi.org/10.1109/ACCESS.2019.2939368
44. Ning J, Cao Z, Dong X, Liang K, Wei L, Choo K-KR (2021) CryptCloud+: secure and expressive data access control for cloud storage. IEEE Trans Serv Comput 14(1):111–124. https://doi.org/10.1109/TSC.2018.2791538
45. Wu Z, Zhang Y, Xu E (2020) Multi-authority revocable access control method based on CP-ABE in NDN. Future Internet 12(1):1–15. https://doi.org/10.3390/fi12010015
46. Fan K, Wang J, Wang X, Yang Y (2017) Proxy-assisted access control scheme of cloud data for smart cities. Pers Ubiquit Comput 21(5):937–947
47. Wang S, Wang X, Zhang Y (2019) A secure cloud storage framework with access control based on blockchain. IEEE Access 7:112713–112725. https://doi.org/10.1109/ACCESS.2019.2929205
48. Saini Q, Zhu N, Singh Y, Xiang LG, Zhang Y (2021) A smart-contract-based access control framework for cloud smart healthcare system. IEEE Internet Things J 8(7):5914–5925. https://doi.org/10.1109/JIOT.2020.3032997
49. Nakamoto S (2008) Bitcoin: A peer-to-peer electronic cash system. https://bitcoin.org/bitcoin.pdf
50. Liebenau J, Elaluf-Calderwood SM (2008) Blockchain innovation beyond bitcoin and banking. In: Legally-Enforceable Fairness in Secure Two-Party Computation Topics in Cryptology—CT-RSA. Springer, pp 121–137
51. Crosby M, Pattanayak P, Verma S, Kalyanaraman V (2016) Blockchain technology: beyond bitcoin. Appl Innov 2:6–10
52. Sankar LS, Sindhu M, Sethumadhavan M (2017) Survey of consensus protocols on blockchain applications. In: Proceeding of the 4th International Conference on Advanced Computing and Communication Systems (ICACCS), pp 1–5
53. Patra TS, Suresh A, Yalame H (2020) ABY2.0: improved mixed-protocol secure two-party computation. Cryptology ePrint Archive, Report. https://ia.cr/2020/1225
54. Buyya R, Ranjan R, Calheiros RN (2009) Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. In: Proceedings of the International Conference on High Performance Computing and Simulation, pp 1–11. https://doi.org/10.1109/HPCSIM.2009.5192685

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.