Computers, Materials & Continua
DOI: 10.32604/cmc.2023.030558
Article

Blockchain Merkle-Tree Ethereum Approach in Enterprise Multitenant Cloud Environment

Pooja Dhiman[1], Santosh Kumar Henge[1], Sartaj Singh[1], Avinash Kaur[2], Parminder Singh[2,3] and Mustapha Hadabou[3,*]

[1]School of Computer Applications, Lovely Professional University, Phagwara, 144001, India
[2]School of Computer Science and Engineering, Lovely Professional University, Phagwara, 144001, India
[3]School of Computer Science, Mohammed VI Polytechnic University, Ben Guerir, 43150, Morocco
*Corresponding Author: Mustapha Hadabou. Email: mustapha.hedabou@um6p.ma
Received: 29 March 2022; Accepted: 12 July 2022

Abstract: This research paper puts emphasis on using cloud computing with Blockchain (BC) to improve the security and privacy in a cloud. The security of data is not guaranteed as there is always a risk of leakage of users' data. Blockchain can be used in a multi-tenant cloud environment (MTCE) to improve the security of data, as it is a decentralized approach. Data is saved in unaltered form. Also, Blockchain is not owned by a single organization. The encryption process can be done using a Homomorphic encryption (HE) algorithm along with hashing technique, hereby allowing computations on encrypted data without the need for decryption. This research paper is composed of four objectives: Analysis of cloud security using Blockchain technology; Exceptional scenario of Blockchain architecture in an enterprise-level MTCE; Implementation of cipher-text policy attribute-based encryption (CP-ABE) algorithm; Implementation of Merkle tree using Ethereum (MTuE) in a Multi-tenant system. Out of these four objectives, the main focus is on the implementation of CP-ABE algorithm. CP-ABE parameters are proposed for different levels of tenants. The levels include inner tenant, outer tenant, Inner-Outer-Tenant, Inner-Outer-External-Tenant, Outer-Inner-Tenant, External-Outer-Inner-Tenant and the parameters such as token, private key, public key, access tree, message, attribute set, node-level, cipher-text, salting which will help in providing better security using CP-ABE algorithm in a multi-tenant environment (MTE) where tenants can be provided with different levels of security and achieved 92 percentage of authenticity and access-control of the data.

Keywords: Blockchain (BC); merkle tree using ethereum (MTuE); multi-tenant environment (MTE); homomorphic encryption (HE); ciphertext policy attribute-based encryption (CP-ABE)

1 Introduction

Security is always a major concern while storing data on the cloud. Since the cloud follows a centralized approach, the data is stored in encrypted form on remote servers. For encryption, different algorithms are used. Most secure technique for encrypting the data considered these days is using a Homomorphic Encryption (HE) algorithm. It allows operations on encrypted data without decrypting it [1]. To provide double security, a block chain can be used with cloud computing and encrypted using a HE algorithm [2]. A Blockchain (BC) is a distributed ledger containing an ordered list of records [3]. BC is an unaltered chain of data connected in a decentralized manner to oppose the cloud's centralized approach. If an attacker anyhow gets access to data stored on the BC, only that piece of data is leaked, the rest data will remain safe as it is stored in different blocks on different remote servers which are owned by different organizations [4]. Decentralization in a cloud can be achieved using Ethereum BC or Hyperledger BC method and Ciphertext Policy Attribute-based Encryption (CP-ABE) algorithm [5]. The homomorphic algorithm can be combined with the CP-ABE mechanism to provide better security. The secure deletion of outsourced data in cloud server storage is one of the most significant concerns in cloud storage environment; the employ of encryption methods are the capable approach to deal with this type of issues. The identity based cryptography is proposing a design to attain secure deletion in cloud storage [6].

The various cloud service providers aim to take action to the client's needs by expenses the minimum potential resources, one of the approaches used by the provider is to pool the resources which are available for allocation among multiple users residing at various distinct multi-tenants [7]. The Side Channel Attacks (SCA) intends at mine confidential and sensitive data from a secure system during quantity and scrutiny of objective factors and shatter cryptography-code by exploiting sensitive data accidentally reveal by secure system. The Zero-Value Point Attack (ZVPA) is a kind of power analysis attack. Sometimes, the SCA has become a grave hazard for cryptographic cloud applications on devices with small resources; indeed, it revolve out that the usual randomization techniques which are cannot stop the recent differential power analysis type attacks which are blended with Refined Power Analysis and ZVPA [8]. The fiber transmission systems (FTS) are the supportive environment to the cloud servers to integrate various cloud services. The space-division multiplexing (SDM) is becoming essential to increase the capacity of future optical communication systems through the cloud services. Some research has demonstrated through experimental measurements the potential of Space-Time coding (STC) in mitigating mode-dependent loss (MDL) in few-mode fiber transmission systems. The Q-factor is improved by 3.4 dB for an MDL of 10 dB [9]. The STC is designed and used for wireless communications which was established to be resourceful to reduce polarization-dependent loss in PolMux systems and MDL in SDM systems [10], these environment are needed to improve the cloud services in enterprise level multi-tenancy. The enterprise level optical transmission systems are fundamental and most priority base service for telecommunication infrastructure for implicating the cloud services among various enterprise level multi-tenants [11]. In this paper, we proposed architecture for a MTCE that is based on using BC for storage and data is encrypted using fully or partially HE techniques. The following Homomorphic types are existing in the present system [12,13]:

- Partial HE: In this type, one operation is allowed at a moment that is either addition or multiplication. Rivest, Shamir, Adleman (RSA) algorithm ($*$), Elgamal ($*$), and Paillier ($+$) come under this category.
- Somewhat HE: More than one operation can be evaluated at the same time but with a limited number of operations. Learning with errors, Ring learning with errors are some examples.

- Fully HE: Any number of operations can be performed on ciphertext by mixed operators, it can be additive, multiplicative, or mixed operation. Brakerski-Gentry-Vaikuntanathan (BGV), Enhanced Homomorphic Cryptosystem (EHC), Non-interactive Exponential Homomorphic Encryption Scheme (NEHE), and Algebra Homomorphic Encryption scheme based on updated ElGamal (AHEE) are fully homomorphic schemes.

The proposed architecture leads to double security and can be used for future systems. In a multi-tenant environment (MTE), BC stores data from multiple tenants after proper verification of data from other nodes and ensures integrity using smart contract technology. The article is organized in the following manner: Section 2 included the research objectives; Section 3 comprises the related works along with the background of the research; Section 4 includes the proposed methodology which has designed based on Implementation of CP-ABE Algorithm. This section has implicated with the Execution Scenario of BC Architecture in Enterprise-Level MTCE and Implementation of Merkle Tree using Ethereum (MTuE) in MTE along with experimental setup; Section 5 contains the experimental analysis, results along with the discussion; Section 6 addresses the conclusion and future direction.

This paper focuses on analyzing past proposed approaches on cloud security using BC to ensure better security and maintain integrity. This research paper has composed with four major objectives:

- Analysis of cloud security using BC technology
- Execution scenario of Enterprise-level MTCE
- Implementation of CP-ABE algorithm
- Implementation of MTuE in a Multi-tenant system

The proposed CP-ABE Multi-tenancy parameters used to improve access control policies, which has implicated with enterprise level dictionary to filter in and out data. It also provides automated key filter mechanism to analyze security and privacy in enterprise level multitenant.

2 Related Works

The complete analysis of past proposed cloud security models and approaches using BC technology in terms of various cloud security issues, techniques, achievements, and lagging issues is described below:

The author Reantongcome et al. [14] has proposed in their paper about multi-tenancy co-resident attack which is caused due to leakage in data by a malicious tenant. The authors implicated a truffle framework to implement blockchain with Ethereum. Techniques used: Ethereum encoded in solidity along with the smart contract is used. Achievements: The transactions between the tenants and the cloud owner are recorded using the BC. Lagging issues: Further work can be done to improve the integrity and the confidentiality of the proposed model. Park et al. [15] find a solution for securing bit-coins by surveying BC technologies. Techniques used: It installs an electronic wallet in the cloud for using the service using a secure bit-coin protocol and after using the service it successfully deletes the user details from the wallet. Public key encryption is used for encrypting the data. Achievements: It verifies users' privacy by completely removing the wallet's details from the cloud. Lagging issues or future scope: Researchers can study the future risks of using bitcoins. Wang et al. [16] proposed a method to improve access control policies. Ethereum blockchain and CP-ABE are used for this purpose. Techniques used: Ethereum blockchain is deployed on Linux/Unix operating system while Windows 10 is used for the implementation process. A smart contract is used to store the ciphertexts. Achievements: Cloud security is improved by decrypting in a valid access period. Accessing the cost

of the files has decreased, making function tracing easier. Lagging issues: To improve data integrity, decentralized technologies can be introduced. Weber et al. [17] evaluate quantitative and qualitative analysis to design a multitenant scalable architecture. The evaluation is based on integrity and isolation in the tenant's performance. Techniques used: Ethereum is used with Laava's industry partner for implementing a proof-of-concept prototype. Achievements: Low cost, data integrity, and performance isolation can be achieved by the proposed architecture. Lagging issues: Flexibility in anchored chains can be evaluated.

Sukhodolskiy et al. [18] ensure privacy in cryptographic operations without the participation of cloud owners. It puts more emphasis on the access control mechanism of the cloud. Techniques used: Cloud access can be controlled using CP-ABE scheme which is implemented on Ethereum. Achievements: It ensures security by storing only hash-based ciphertexts. Shah et al. [19] used IPFS (Inter-Planetary File System) protocol to emphasize data utilization, decentralized data storage, security, and privacy in their paper "Decentralized Cloud Storage Using BC". Techniques used: It stores AES encrypted files on peer networks using IPFS protocol in the BC guarded by smart contracts. Crypto-currency is transferred to the peer's wallet from the user's wallet. Achievements: The block chain's decentralized approach discussed in this paper is considered to be safe and secure. Qu et al. [20] proposed a technology for electronic voting in their paper which focuses on security and privacy issues by combining BC and homomorphic signcryption. Techniques used: Homomorphic signcryption technique in Hyperledger blockchain type is used to secure the voting ballots. Valid and invalid votes after aggregating the voting results are categorized using a smart contract. Achievements: It improves and secures the voting process by reducing the total time required in a secure and transparent manner without the need of any third party. Lagging issues: Security can be further improved by practically implementing the proposed model and researching the model in detail. Sharath Yaji et al. proposed a technique using partial HE (PHE) schemes with BC in their paper to preserve privacy. It focuses on attacks on the wallet, collision attack reimages attacks in the BC. Techniques used: Goldwasser-Micali and Paillier PHE schemes are used for the proposed model which can bypass most of the attacks. Achievements: The time required to process the model is comparatively less and is more secure. Lagging issues: Latest attacks can be analyzed and experimented on in the proposed model with improved PHE schemes.

Kumar et al. [21] concluded a privacy algorithm to focus on malicious attacks, in the paper "BMIAE: BC-based multi-instance Iris authentication using additive ElGamal HE" which is based on BC multi-instance iris authentication using ElGamal HE (BMIAE). It is mainly designed for an un-trusted server. Techniques used: Distance compute can be performed using a smart contract in BMIAE, where Elgamal is based on the hardness of these discrete problems. Achievements: The proposed model guarantees confidentiality and integrity can be achieved along with the decreased execution time and the computational cost. Huang et al. [22] Proposed BPS which is the Blockchain-Publish-Subscribe model used to secure pub/sub streaming models from the attacks on the edge cloud caused by multiple tenants using the same pub/sub system. Techniques used: The illegal or unauthorized behavior by a tenant or an unauthorized user can be detected using the BPS model which is based on BC functionality. It verifies the integrity using a Merkel tree and keeps a record of all the tenants in the access control list combined with the smart storage feature. Achievements: It is analyzed that the proposed model outperforms security with a minimum overhead of performance. Lagging issues: BPS can be tested to support complex deployments and improve scalability. Ismail et al. [23] analyzed in their paper a new paradigm combining cloud-based services with blockchain technology to provide an efficient and patient-centric view to the healthcare stakeholders. Techniques used: A Blockchain-cloud integration (BcCI) is deployed for managing the patient details efficiently in the

database management system for the healthcare industry. Further, this paper discusses the strengths and weaknesses of BcC architecture. Achievements: BcCI can overcome the individual shortcomings of cloud and blockchain technologies; as the cloud provides a centralized service that may violate the privacy of the patient and blockchain is not scalable and it faces some challenges with its efficiency. Lagging issues: Future scope is to enhance the existing model for better scalability and efficiency. Xie et al. [24] integrated BC technology with the cloud data integrity verification scheme. Technology used: The authors introduced a lattice signature algorithm to resist quantum computing technology and combined it with a cuckoo filter for simplifying overhead in the computational process. It relies on a small integer solution (SIS) assumption to cope with the attacks. Achievements: The proposed scheme can cope with quantum attacks and malicious attacks with high efficiency. Lagging issues: To explore more features of the data integrity scheme combined with BC technology.

Yang et al. [25] designed a blockchain-based access control framework by integrating cloud computing. The access control permission of the data on the cloud is redefined, which is stored in the BC. The proposed model overcomes the limitations of the BC and the cloud. Techniques used: AuthPrivacyChain is the framework designed by the authors based on EOS BC. The proposed model is compared with the traditional cloud using the test tool JMeter. Achievements: The users with proper access rights can only access the data. The designed model can prevent the data from insider as well as outsider attacks. Li et al. [26] surveyed blockchain-based trust models. Blockchain's decentralized approach helps in protecting the data from breaches as the traditional cloud trust model is not transparent and follows a centralized approach that cannot be trusted. Techniques used: In cloud computing, the efficiency and adaptiveness can be improved by using a hybrid edge cloud with double-BC technology. Achievements: BC successfully builds trust by using a transparent approach, it avoids data leakage and eliminates the single point of failure because of its decentralized approach. Lagging issues/Future scope: In the future, BC must be studied to collaborate with the new cloud technologies like edge computing, Internet of Things (IoT) applications, fog computing, and so on. The resource constraint may create a problem if all the data is stored on the chain, it may increase the processing time. Hence, it is to be researched. Uddin et al. [27] reviewed the common cloud vulnerabilities that are mostly based on the virtualization platform, the identified issues are solved using the BC-enabled models. Techniques used: The common vulnerabilities like a centralized security risk, transparency, resource sharing in a virtualized environment are discussed. Achievements: BC helps the cloud service providers in creating a virtual database and using a one-click method for accessing the services. Ingole et al. [28] analyzed the business opportunities in both financial and non-financial sectors using BC technology. Techniques used: The processing of bit coin is discussed; attackers may steal the private key stored in the user's computer hard drive to hack bit coin. Achievements: The authors of this paper suggested a way to secure the data from breach by simply removing the user's information from the system completely. Alharbi et al. [29] introduced a literature survey related to the HE algorithm. Techniques used: The survey aims to reduce the gap between the algorithm and its applications in terms of providing security. The homomorphic applications like vehicle communication, electronic voting, cloud computing, BC, and signal processing are discussed in the paper. Achievements: The homomorphic algorithm is considered as providing a high level of security by allowing the operations on encrypting data. Lagging issues: In the future, a complete systematic view of the homomorphic algorithm can be analyzed. Gafif et al. [30] proposed in their paper ciphertext-policy attribute-based encryption (CP-ABE) key encapsulation mechanism. This scheme reduces the expense for the encryption operation as in traditional CP-ABE. Techniques used: The authors proposed two CP-ABE mechanisms, one for the untrusted ABE service provider and another for the semi-trusted ABE service provider. Achievements: The proposed schemes are found

to be secured under CPA and are considered as more efficient than the traditional CP-ABE scheme. HE technique is the latest and most secure way of protecting cloud data on the server. It allows computational operations on encrypted data as there is no need to decrypt the data for processing it [31]. The distributed structure of BC is naturally suitable for the Industrial IoT, which can be used to build distributed trustworthy Industrial IoT with high security [32]. The BC is based on a distributed network and has the characteristics of non-tampering and traceability of block data. It is thus naturally able to solve the security problems of the sensor networks [33]. BC technology has the characteristics of decentralization, openness, transparency, reliability and non-tampering [34]. BC can record historical data by establishing a collectively maintained and tamper-resistant public ledger to ensure the security and reliability of the data stored in a distributed network [35]. In a blockchain system, the node does not rely on any central organization, and every node keeps an entire copy of the transaction database [36]. A BC-based authentication and key agreement protocol is designed for the multi-TA network model, moving the computing load of TA down to the RSU to improve the efficiency of authentication [37]. In a multi-server environment, each server is usually independent of each other [38]. Hashing has become a promising technique to be applied to the large-scale visual retrieval tasks. Multi-view data has multiple views, providing more comprehensive information [39].

The existing approaches and modals are majorly focused on authentication and authorization encryption and decryption techniques with the integration of limited key-filter services with bondable network architecture and single enterprise tenants. The proposed CP-ABE parameters are integrated for different levels of tenants. The levels include inner tenant, outer tenant, Inner-Outer-Tenant, Inner-Outer-External-Tenant, Outer-Inner-Tenant, External-Outer-Inner-Tenant and the parameters such as token, private key, public key, access tree, message, attribute set, node-level, cipher-text, salting which will help in providing better security using CP-ABE algorithm in a MTE where tenants can be provided with different levels of security.

3 Implementation of Ciphertext Policy Attribute-Based Encryption (CP-ABE) Algorithm in Multi-Tenancy Environment

The CP-ABE algorithm consists of four algorithms namely, Setup, Encrypt, KeyGen, and Decrypt [40,41].

Setup: The public key and master key are generated using the bilinear group with a prime number and two random generators. The input is based on security parameters and the universe of the attribute. It takes no other input than the implicit parameters.

Encrypt (Public_key_PK, Message_M, Access_tree_T): Input values are a public key, message, and a tree-like access structure. The message can be encrypted using an access structure like a tree. A polynomial is chosen for each node in a top-down manner in such a way that, the degree of each node is one less than the threshold value. The ciphertext can only be accessed if the user satisfies the attribute set for the access tree.

KeyGen (Master_key, Attribute_set_S): It generates the private key by taking as input the master key and attribute set S, describing the key.

Decrypt (Cipher_text_CT, private_key_SK, node_N): Original message can be decrypted using a combination of public and private keys, access tree structure, and ciphertext. Based on the condition of satisfying the set of attributes S by the access tree, the ciphertext can be decrypted.

In Fig. 1, multi-tenant architecture is created where Sender Tenant (ST) sends a message to Receiver Tenant (RT) by encrypting it using the CP-ABE algorithm. HE can be applied for double

security. A HE algorithm can be used in combination with the CP-ABE algorithm to make it double secured. It allows any number of operations on the ciphertext itself.

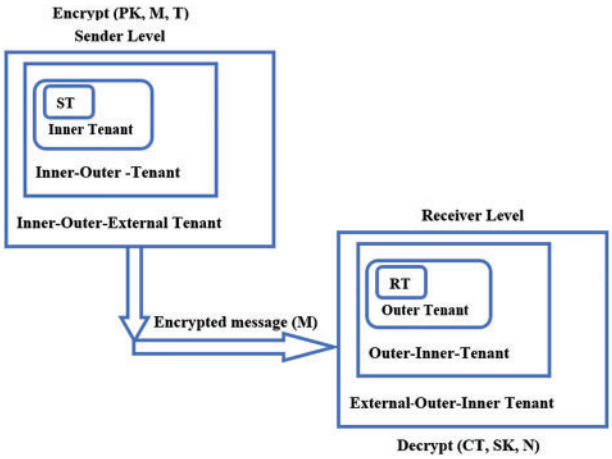

Figure 1: Ciphertext Policy Attribute-based Encryption (CP-ABE) algorithm in multiple tenants

3.1 Execution Scenario of Blockchain Architecture in Enterprise-Level Multi-Tenant Cloud Environment

Individual permission BC for each tenant is used to maintain data in a MTE. The data is stored in the form of blocks in the BC and hashing method is used for the encryption of the data. These blocks are connected using the hash value of the previous block in chain form. New data is inserted in the BC only after successful authentication. The implementation of the Merkel tree is done using Ethereum's Merkle-Patricia library. Tenants can rely on the cloud service provider for the anchoring process. The communication with the anchoring component is made using a trigger. It writes content to the BC [42]. Blockchain uses a consensus algorithm for this purpose. A new node is inserted only after validation from the anchored node in the Merkel tree. It will check if data already exists in the BC. The other nodes in the BC network must permit the inclusion of data. The BC header and nonce measure cryptographic secure hash function SHA-256. HE can be used for the encryption process while transmitting data from one node to another in a BC along with the hash function. All the transactions are recorded in a Merkel tree using hashing or anchoring method. Trust and data integrity among tenants can be maintained using smart contract technology. It is a self-execution program based on some conditions. Each tenant executes its customized chain of block chains without interrupting other tenants. BC implements a decentralized consensus protocol that guarantees trustworthy transactions in a non-secure environment [43]. The software retailers structured SaSE implicated applications through a MTE is not at all an easy undertaking. It has many advantages along with some lagging issues.

The MTE testing is one of the extremely inspiring technical fields. The MTE-test-cases are principally attentions over various supporting networks, service environment (SE), infrastructures, applications and technical components. MTE-test-cases are implicated based on various supporting service layers such as Application as SE (AaSE), Software as SE (SaSE), Infrastructure as SE (IaSE), and Network as SE (NaSE) as shown in Fig. 2. The layer1 integrated by SaSE with supporting sub-levels of execution among various vendors, customers and feasible applications. The layer2 integrated by AaSE both functional and non-functional testing scenarios. The layer3 integrated by NaSE with testing of the network which is required to be carried out from the security perspective, flow of data

and encryption/decryption techniques such as Secure Sockets Layer (SSL). The layer4 integrated by IaSE with testable MTE applications over the infrastructures.

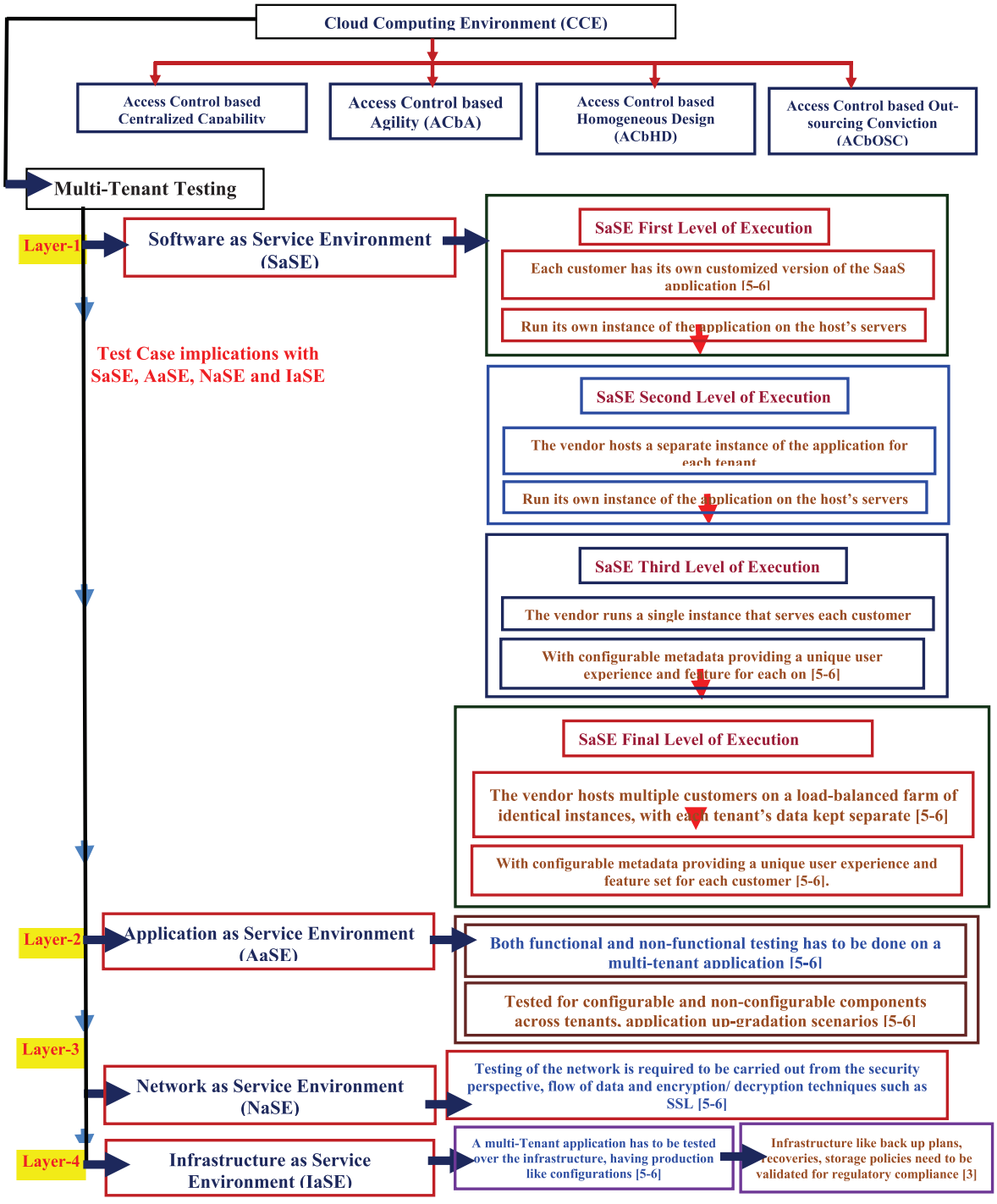

Figure 2: Multi-tenant testing with Service Environment (SE) through Application as SE (AaSE), Software as SE (SaSE), Infrastructure as SE (IaSE) and Network as SE (NaSE)

3.2 Implementation of Merkle Tree Using Ethereum (MTuE) in Enterprise-Level Multi-Tenant Environment (MTE)

Fully HEA allows computations on encrypted data without releasing the decryption key to the cloud security and privacy [44]. Merkle tree is an essential part of a BC. Only leaf nodes store the data while non-leaf nodes only store the hash of the previous block [18,40,45]. The Merkle tree used in the Ethereum BC is known as the Merkle Patricia tree, which has three different structures: State root, Transaction root, and Receipt root as shown in Fig. 3. Each node N in a Merkel tree can be evaluated using the expression below:

$$\text{node} = \text{hash}\ \Sigma\ (\text{node.children}\ (\text{N})\ .\text{hash}) \tag{1}$$

Here, a node is integrated with the help of encrypting hash table by incorporating the local node. Hash is a key that can be either a token key or salting or a public key. Hence, by evaluating the formula for calculating node, it can be stated as it is a combination of all the hash values and its children can be evaluated by using a hash table. Here, a node is a communicating device and children are tenants which belong to a user or an organization. It authenticates data using hashing technique in a MTE. Each node except the leaf node contains the hash of their child nodes (child nodes taken in pairs). If a leaf node is changed, then the hash of all the nodes in the path from the root node to the changed leaf node will change [46]. The key-value pair in the state tree is updated in the Merkel tree after every transaction. The smart contract ensures data integrity.

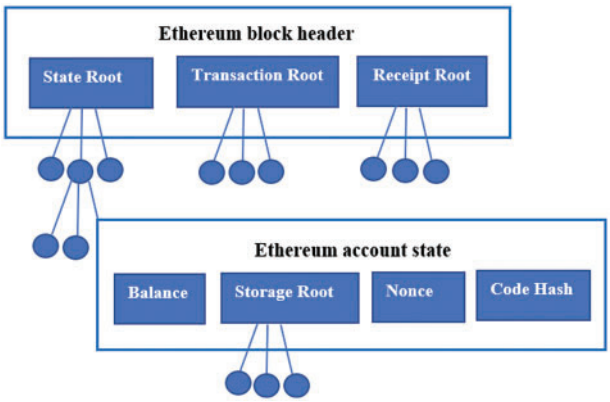

Figure 3: Implementation of merkel tree using ethereum

The account address is stored in the key and the value is encoding of the Ethereum account state that is, Balance, Storage root, Nonce, Code hash. A block contains 17 elements out of which, 15 elements are part of the block header. It contains Parent hash, State Root, Transactions Root, Receipts Root, Number, Gas Limit, Timestamp, Mix hash, Nonce. Nonce verifies that computations are successfully implemented on some particular block. It is an 8-byte hash. A state root gives the value of the hash of the root node after some transaction is finalized. Transaction Root gives the hash value of the root node populated from a transaction's list of a block. Receipts Root is the hash value of the root node populated from the transaction's receipt list for each transaction. The account state stores information on some particular account. It has four variables. Nonce gives the number of transactions made on the account. Balance is the amount owned by the user, which is stored in the form of key-value pair. Storage Root is a 32 byte or 256-bit hash code of root node which is used to encode the contents of the account. Code Hash is the hash of the Ethereum Virtual Machine of the account's contract.

These hash values cannot be modified and are executed only upon a message call on that address. The node acts as a communicating device between the different levels of tenants or children. A group of machines or users or an organization can be considered children in this architecture. Various levels of tenants communicate using these nodes. A combination of any of the tenant levels from the Internal-External-Global setup is used. Packets are transmitted via the nodes to the destination children which means, communicating device or number of nodes will be decided dynamically according to the architecture requirement. Parameters are then applied according to the level of tenant communication and according to the number of nodes and children setup is implemented. A partial experiment is prepared to implement the proposed model.

3.3 Experimental Setup and Test Cases

In the proposed architecture, CP-ABE algorithm is integrated with block-chain technique as shown in Fig. 4. Different layers of tenants are customized and a permission block-chain is used for the storage. A total of 120 systems are integrated with 5 groups of multi-tenants which each group contain 5 users or tenants. Comparative analysis of CP-ABE based merkle tree with ethereum enterprise level multitenancy is shown in Fig. 5 as per the experiment setup in Tab. 1.

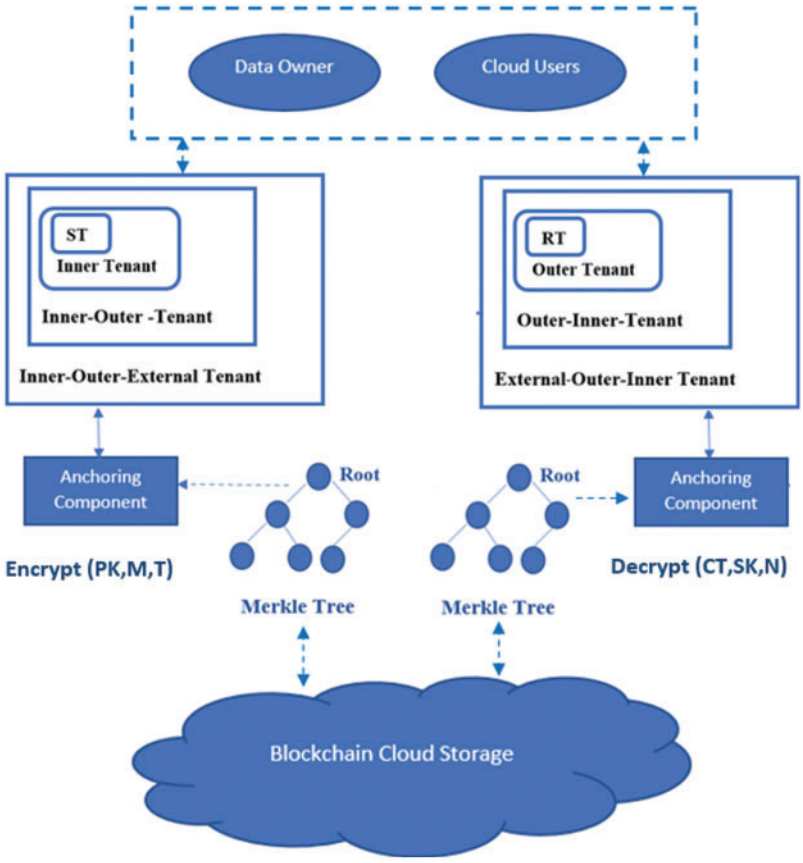

Figure 4: Integrating CP-ABE and block-chain in a multi-tenant environment

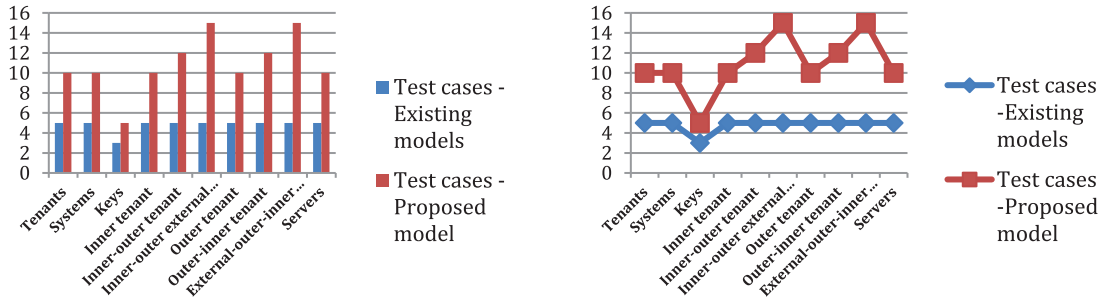

Figure 5: Comparative analysis of CP-ABE based merkle tree with ethereum enterprise level multi-tenancy experimental setup

Table 1: CP-ABE based Merkle Tree with Ethereum (MTuE) enterprise level multi-tenancy experimental setup

Number of #	CP-ABE based MTuE enterprise level MTE
Tenants	10
Systems	10
Keys	5
Inner tenant	10
Inner-outer tenant	12
Inner-outer external tenant	15
Outer tenant	10
Outer-inner tenant	12
External-outer-inner tenant	15
Servers	10

The Tab. 2 describes the parameter generation at various levels in a CP-ABE algorithm. The terms used in the Tab. 2 are T-Access tree, PK–Public key, SK–Secret key or private key, M–Message, CT–ciphertext, N–Node in a tree, For Internal-Global and Global-Internal tenant setup, token key 'Token' is used. Since for internal communication, only token keys can be used with no need of using an extra combination of keys which is required for global communication. An additional key with the token key is sent along with the packets in such communications. For encryption, token, salting and public keys are used in sender enterprise to encrypt the packets. In key generation, a secure key is added to send packets to another enterprise tenant and a combination of parameters like token, salting, public key, private key or secret key, ciphertext, and N-based are used. For Internal-External and External-Internal tenant setup, communication is done locally so an additional key is not required here. Token key is sufficient to provide security.

Table 2: CP-ABE-multi-tenancy parameters

Multitenant/CP-ABE	Setup	Parameters	Encrypt	Key gen	Decrypt
Sender tenant	Internal-global	Token, PK, M, T	Token, PK, M, T	PK, M, T	CT, SK, N, Token
Inner-tenant	Internal-external	Token-based	Token-based	Token based	Token based
Outer tenant	External-global	Token + Salting based	Token + Salting based	Token + Salting + CT based	Salting + Token + CT based
Inner-outer-tenant	Internal-external-global	Token + Salting + PK based	Token + Salting + PK based	Token + Salting + PK + M + T + CT	Token + Salting + CT + SK based
Inner-outer-external tenant	Internal-external-global-global	Token + Salting + PK + M + T	Token + Salting + PK + M + T based	Token + Salting + PK + M + T + CT + SK + N based	Token + Salting + CT + SK + N based
External-outer-inner tenant	Global-global-external-internal	T + M + PK + Salting + Token	T + M + PK + Salting + Token	N + SK + CT + T + M + PK + Salting + Token	N + SK + CT + Salting + Token based
Outer-inner-tenant	Global-external internal	PK + Salting + Token based	PK + Salting + Token based	CT + T + M + PK + Salting + Token	SK + CT + Salting + Token
Outer tenant	Global-external	Salting + Token based	Salting + Token based	CT + Salting + Token based	CT + Token + Salting based
Inner-tenant	External-internal	Token-based	Token-based	Token-based	Token based
Receiver tenant	Global-internal	PK, M, T, Token	CT, SK, N	PK, M, T	Token, CT, SK, N,

4 Results and Discussion

Multiple tenants are created as an inner tenant, outer tenant, Inner-Outer-Tenant, Inner-Outer-External Tenant, Outer-Inner-Tenant, External-Outer-Inner Tenant. The proposed methodology has experimented and executed with eight stages test cases as shown in the Fig. 1. Each test case has integrated with set-of enterprise level directory based access rules. These rules are performed has performed on checkpoint console device by using Gaia operating system. The rules are framed and executed based on type of user, authenticity, and access-control, type of environment, local-set of user and browser level patches.

Stage-1: Sender tenant: The sender tenant is the user or organization that wants to send a packet(s) to another tenant at the enterprise level. A sender tenant uses a token along with the public key for encrypting the nodes on the access tree. It acts on the Internal-Global setup.

Stage-2: Inner tenant: It is a tenant which lies in the sender tenant block. Internal-external setup is required for this level. It is completely based on token only. No keys are required for internal communication among tenants.

Stage-3: Inner-outer tenant: It is based on an Internal-External-Global setup. It uses token, salting and public key for encrypting the message. For key generation access tree (collection of nodes or tenants) is used with message and ciphertext along with the token, salting and public key. The decryption of ciphertext can be done using a secret key with salting and token.

Stage-4: Inner-Outer-External Tenant: It is based on the Internal-External-Global-Global setup. Here, the packets are transmitted outside the enterprise. For encryption, token, salting, public key, message and access tree are used. A combination of public and private keys is generated in this setup. Decryption is done based on nodes, and token, salting and a private key are used on the ciphertext for decryption.

Stage-5: External-Outer-Inner Tenant: It is based on a Global-Global-External-Internal tenant setup. It is responsible to receive the packets from the sender enterprise tenant. Encryption of message or packet on access tree is done using salting, a public key and a token key.

Stage-6: Outer-Inner-Tenant: It is based on the Global-External-Internal setup. It takes the packets from the outermost layer of the tenants and sends them to the outer tenant layer which is next to the receiving tenant. A public key, salting and token key are used for the encryption process, and for the decryption process, a private key, salting and token key are used.

Stage-7: Outer tenant: In this level, a packet can be received and decrypted using the salting technique, which adds some random collection of digits, symbols and alphabets in the password field to make it difficult to guess by a hacker or an intruder. It used salting, secret key, or private key and token to decrypt the ciphertext received. It acts on an External-Global setup.

Stage-8: Receiver tenant: The tenant for whom the packet is meant can receive the packet and ciphertext can be decrypted using a combination of private key and token on the nodes.

For all other tenant setups, salting is used along with the token keys to provide an extra layer of protection that is multi-factor authentication mechanism is used here. For example, some financial institutions are already using such secured systems which require multiple levels of authentication from the user. It requires a combination of username, password, one-time password (OTP), and salting technique combined with encryption, or an extra level of authentication added for better security. The additional set of keys results in better security when communicating at a global level tenant mechanism. The proposed CP-ABE based Merkle Tree with Ethereum Enterprise level Multi-tenancy model has compared and evaluated with existing models based on the integration of number of tenants, systems, keys, inner-tenants, Inner-outer tenants, Inner-outer external tenants, Outer tenants, Outer-inner tenant along with External-outer-inner tenants as shown in the Tab. 3, and achieved good success rate in data-access-control, authentication, authorization, key generation, key-passing between the tenants, self-mutation and data-transmission among the tenants of same enterprise and among diverse enterprise tenants. The Tab. 4 is representing the cross verification success rate among the number of nodes in the dissimilar environments such as Tenants, Systems, Keys, Inner tenant, Inner-outer tenant, Inner-outer external tenant, Outer tenant, Outer-inner tenant, and External-outer-inner tenant along with the supporting interfacing servers.

Table 3: CP-ABE MTuE enterprise tenant experimental setup with nodes, existing and proposed models

Number of #	Tenants	Systems	Keys	Inner tenant	Inner-outer tenant	Inner-outer external tenant	Outer tenant	Outer-inner tenant	External-outer-inner tenant	Servers
Test cases-existing models	5	5	5	5	5	5	5	5	5	5

(Continued)

Table 3: Continued

Number of #	Tenants	Systems	Keys	Inner tenant	Inner-outer tenant	Inner-outer external tenant	Outer tenant	Outer-inner tenant	External-outer-inner tenant	Servers
Test cases-proposed model	10	10	5	10	12	15	10	12	15	10

Table 4: CP-ABE based MTuE enterprise level MTE experimental setup with number of nodes *vs.* data access control, data-transmission (DS) and types of tenants

Number of #/ success rate in percentage	Data-access-control	Authentication	Authorization	Key generation	Key-passing	Self-mutation	DS tenants of same enterprise	DS diverse enterprise tenants
Tenants	98.5	96	96.5	91	93	89	93	93.5
Systems	97.5	97	98.5	96	94.5	92	95	94.5
Keys	95.5	93	96.5	94.3	93.5	96	97.5	95.5
Inner tenant	96.5	94.5	93.5	95.3	92.5	94.5	96.5	94.5
Inner-outer tenant	92.5	90.5	91.5	90.3	89.5	90.5	89.5	90.5
Inner-outer external tenant	**87.5**	**86.3**	**89.5**	**91.5**	**89**	**91.5**	**90.5**	**91.5**
Outer tenant	96.5	94.5	93.5	95.3	92.5	94.5	96.5	94.5
Outer-inner tenant	92.5	90.5	91.5	90.3	89.5	90.5	89.5	90.5
External-outer-inner tenant	87.5	86.3	89.5	91.5	89	91.5	90.5	91.5
Servers	90.5	92.5	93.5	92.5	92.5	91	92.5	91.5

The cross verification success rate has analyzed and computed based on the node function as shown in Eq. (1) by integration of various supporting key parameters such as data-access-control, authentication, authorization, key generation, key-passing between the tenants, self-mutation and data-transmission among the tenants of same enterprise and among diverse enterprise tenants. The cross verification success rate will be differ enterprise to enterprise and environment to environment or regions. This research has achieved 90% to 91.5% of cross verification success rate with the integration of localized-cloud environment.

5 Conclusion

The Block-chain technology can be implemented on a cloud for storage purposes using Ethereum or Hyperledger methods and the CP-ABE algorithm. Data can be encrypted using the Homomorphic algorithm technique for providing better security. This research paper has implicated with analysis of cloud security using BC technology in an enterprise-level multi-tenant cloud environment with

technical feasibility of MTuE into the consideration of technical parameters of CP-ABE, which has introduced where multiple tenants are created as various levels of tenants like an inner tenant, outer tenant, Inner-Outer-Tenant, Inner-Outer-External Tenant, Outer-Inner-Tenant, External-Outer-Inner Tenant. An additional set of keys is generated using a HE algorithm with ciphertext policy which results in a more secured system. The proposed model is considered to be double secured by using a combination of HE and an additional set of keys generated globally for sharing packets at an enterprise level. A partial experiment has been prepared using parameter categorization at different levels of tenants. These parameters are token, private key, public key, access tree, messages; attribute set, node-level, ciphertext and salting. The proposed CP-ABE algorithm helps in providing better security in a MTE where tenants can be provided with different levels of security and achieved 92 percentage of authenticity and access-control of the data. The scope of the further execution stage of the proposed paradigm is planning to extend with auto key-filter authentication and authorization mechanism to control the data-access among the multi-tenant of single enterprise and multi-tenant of different enterprises.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

References

[1] S. Yaji, K. Bangera and B. Neelima, "Privacy preserving in blockchain based on partial homomorphic encryption system for AI applications," in *Proc. of IEEE 25th Int. Conf. on High Performance Computing Workshops,* Bengaluru, India, pp. 81–85, 2018.

[2] X. Yan, Q. Wu and Y. Sun, "A homomorphic encryption and privacy protection method based on blockchain and edge computing," *Wireless Communications and Mobile Computing,* vol. 2020, pp. 1–9, 2020.

[3] A. Gorkhali, L. Li and A. Shrestha, "Blockchain: A literature review," *Journal of Management Analytics,* vol. 7, no. 3, pp. 321–343, 2020.

[4] C. H. V. N. U. B. Murthy, M. L. Shri, S. Kadry and S. Lim, "Blockchain based cloud computing: Architecture and research challenges," *IEEE Access,* vol. 8, pp. 205190–205205, 2020.

[5] N. Helil and K. Rahman, "CP-ABE access control scheme for sensitive data set constraint with hidden access policy and constraint policy," *Security and Communication Networks,* vol. 2017, pp. 1–13, 2017.

[6] A. Bentajer and M. Hedabou, "An IBE-based design for assured deletion in cloud storage," *Journal of Cryptologia,* vol. 141, pp. 559–564, 2019.

[7] A. Azougaghe, M. Hedabou, O. Oualhaj, M. Belkasmi and A. Kobbane, "Many-to-one matching game towards secure virtual machine migrating in cloud computing," in *Proc. of Int. Conf. on Advanced Communication System and Information Security,* Marrakesh, Morocco, 2016.

[8] M. Hedabou, "Some ways to secure elliptic curves cryptosystems," *Advances in Applied Clifford Algebras,* vol. 18, pp. 677–688, 2008.

[9] A. El-Mehdi, G. R. B. Othman, L. Bigot, M. Song, E. R. Andresen *et al.,* "Experimental demonstration of space-time coding for MDL mitigation in few-mode fiber transmission systems," in *Proc. of IEEE European Conf. on Optical Communication,* Gothenburg, Sweden, 2017.

[10] E. M. Amhoud, G. R. B. Othman and Y. Jaouën, "Concatenation of space-time coding and FEC for few-mode fiber systems," *IEEE Photonics Technology Letters,* vol. 29, no. 7, pp. 603–606, 2017.

[11] E. M. Amhoud, G. R. B. Othman and Y. Jaouën, "Design criterion of space-time codes for SDM optical fiber systems," in *Proc. of IEEE 23rd Int. Conf. on Telecommunication,* Thessaloniki, Greece, pp. 1–5, 2016.

[12] S. Gaikwad and R. A. Buchade, "Survey on securing data using homomorphic encryption in cloud computing," *International Journal of Computer Sciences and Engineering,* vol. 4, no. 1, pp. 17–21, 2016.
[13] B. Chen and N. A. Zhao, "Fully homomorphic encryption application in cloud computing," in *Proc. of IEEE 11th Int. Computer Conf. on Wavelet Active Media Technology and Information Processing,* China, 2014.
[14] V. Reantongcome, V. Visoottiviseth, W. Sawangphol, A. Khurat, S. Kashihara *et al.,* "Securing and trustworthy blockchain-based multi-tenant cloud computing," in *Proc. of IEEE 10th Symp. on Computer Applications and Industrial Electronics,* Malaysia, pp. 256–261, 2020.
[15] J. H. Park and J. H. Park, "Blockchain security in cloud computing: Use cases, challenges, and solutions," *Symmetry,* vol. 9, no. 8, pp. 1–13, 2017.
[16] S. Wang, X. Wang and Y. Zhang, "A secure cloud storage framework with access control based on blockchain," *IEEE Access,* vol. 7, pp. 112713–112725, 2019.
[17] I. Weber, Q. Lu, A. B. Tran, A. Deshmukh, M. Gorski *et al.,* "A platform architecture for multi-tenant blockchain-based systems," in *Proc. of IEEE Int. Conf. on Software Architecture,* Germany, pp. 101–110, 2019.
[18] I. Sukhodolskiy and S. Zapechnikov, "A blockchain-based access control system for cloud storage," in *Proc. of IEEE Conf. of Russian Young Researchers in Electrical and Electronic Engineering,* Russia, pp. 1575–1578, 2018.
[19] M. Shah, M. Shaikh, V. Mishra and G. Tuscano, "Decentralized cloud storage using blockchain," in *Proc. of 4th Int. Conf. on Trends in Electronics and Informatics,* India, pp. 384–389, 2020.
[20] W. Qu, L. Wu, W. Wang, Z. Liu and H. Wang, "A electronic voting protocol based on blockchain and homomorphic signcryption," *Concurrency and Computation: Practice and Experience,* vol. 34, no. 16, pp. 1–17, 2020.
[21] M. M. Kumar, M. V. N. K. Prasad and U. S. N. Raju, "BMIAE: Blockchain-based multi-instance iris authentication using additive ElGamal homomorphic encryption," *IET Biometrics,* vol. 9, no. 4, pp. 165–177, 2020.
[22] B. Huang, R. Zhang and Z. Lu, "BPS: A reliable and efficient pub/sub communication model with blockchain-enhanced paradigm in multi-tenant edge cloud," *Journal of Parallel and Distributed Computing,* vol. 143, pp. 167–178, 2020.
[23] L. Ismail, H. Materwala and A. Hennebelle, "A scoping review of integrated blockchain-cloud (Bcc) architecture for healthcare: Applications, challenges and solutions," *Sensors,* vol. 21, no. 11, pp. 3753, 2021.
[24] G. Xie, Y. Liu, G. Xin and Q. Yang, "Blockchain-based cloud data integrity verification scheme with high efficiency," *Security and Communication Networks,* vol. 2021, pp. 1–15, 2021.
[25] C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao *et al.,* "Authprivacychain: A blockchain-based access control framework with privacy protection in cloud," *IEEE Access, Special Section on Blockchain Technology: Principles and Applications,* vol. 8, pp. 70604–70615, 2020.
[26] W. Li, J. Wu, J. Cao, N. Chen, Q. Zhang *et al.,* "Blockchain-based trust management in cloud computing systems: A taxonomy, review and future directions," *Journal of Cloud Computing,* vol. 10, no. 1, pp. 1–34, 2021.
[27] M. Uddin, A. Khalique, A. K. Jumani, S. S. Ullah and S. Hussain, "Next-generation blockchain-enabled virtualized cloud security solutions: Review and open challenges," *Electronics,* vol. 10, no. 20, pp. 1–23, 2021.
[28] K. R. Ingole and S. Yamde, "Blockchain technology in cloud computing: A systematic review," *International Research Journal of Engineering and Technology,* vol. 5, no. 4, pp. 1–43, 2018.
[29] A. Alharbi, H. Zamzami and E. Samkri, "Survey on homomorphic encryption and address of new trend," *International Journal of Advanced Computer Science and Applications,* vol. 11, no. 7, pp. 618–626, 2020.
[30] H. E. Gafif and A. Toumanari, "Efficient ciphertext-policy attribute-based encryption constructions with outsourced encryption and decryption," *Security and Communication Networks,* vol. 2021, pp. 1–17, 2021.

[31] P. Dhiman and S. K. Henge, "Analysis of blockchain secure models and approaches based on various services in multi-tenant environment," *Recent Innovations in Computing Lecture Notes in Electrical Engineering,* vol. 855, pp. 563–571, 2022.
[32] J. Wang, B. Wei, J. Zhang, X. Yu and P. K. Sharma, "An optimized transaction verification method for trustworthy blockchain-enabled IIoT," *AdHoc Networks,* vol. 119, pp. 102526, 2021.
[33] J. Wang, W. Chen, L. Wang, R. S. Sherratt, O. Alfarraj *et al.,* "Data secure storage mechanism of sensor networks based on blockchain," *Computers, Materials & Continua,* vol. 65, no. 3, pp. 2365–2384, 2020.
[34] J. Wang, W. Chen, Y. Ren, O. Alfarraj and L. Wang, "Blockchain based data storage mechanism in cyber physical system," *Journal of Internet Technology,* vol. 21, no. 6, pp. 1681–1689, 2020.
[35] J. Y. Zhang, S. Q. Zhong, T. Wang, H. C. Chao and J. Wang, "Blockchain-based systems and applications: A survey," *Journal of Internet Technology,* vol. 21, no. 1, pp. 1–14, 2020.
[36] J. Zhang, S. Zhong, J. Wang, X. Yu and O. Alfarraj, "A storage optimization scheme for blockchain transaction databases," *Computer Systems Science and Engineering,* vol. 36, no. 3, pp. 521–535, 2021.
[37] Z. Xu, W. Liang, K. C. Li, J. Xu and H. Jin, "A blockchain-based roadside unit-assisted authentication and key agreement protocol for internet of vehicles," *Journal of Parallel and Distributed Computing,* vol. 149, pp. 29–39, 2021.
[38] Z. Iggaramen and M. Hedabou, "FADETPM: Novel approach of file assured deletion based on trusted platform module," in *Proc. Int. Conf. of Cloud Computing Technologies and Applications,* Rabat, Morocco, pp. 49–59, 2017.
[39] L. Y. Xiang, X. B. Shen, J. H. Qin and W. Hao, "Discrete multi-graph hashing for large-scale visual search," *Neural Processing Letters,* vol. 49, no. 3, pp. 1055–1069, 2018.
[40] J. Bethencourt, A. Sahai and B. Waters, "Ciphertext-policy attribute-based encryption," in *Proc. of IEEE Symp. Security Privacy,* Berkeley, USA, pp. 321–334, 2007.
[41] Z. Liu, J. Xu, Y. Liu and B. Wang, "Updatable ciphertext-policy attribute-based encryption scheme with traceability and revocability," *IEEE Access,* vol. 7, pp. 66832–66844, 2019.
[42] M. Dameron, "Beigepaper: An ethereum technical specification," *Memory and Storage Beigepaper,* vol. 0.7.2, pp. 1–25, 2018. https://github.com/chronaeon/beigepaper/blob/master/beigepaper.pdf.
[43] S. Xie, Z. Zheng, W. Chen, J. Wu, H. N. Dai *et al.,* "Blockchain for cloud exchange: A survey," *Computers & Electrical Engineering,* vol. 81, no. 2020, pp. 1–12, 2020.
[44] P. Dhiman and S. K. Henge, "Comparative analysis of cloud security complexities and past proposed non-homomorphic and homomorphic encryption methodologies with limitations," *ICT for Competitive Strategies-CRC Press,* vol. 1, pp. 787–799, 2020.
[45] A. S. Yahaya, N. Javaid, R. Khalid, M. Imran and N. Naseer, "A blockchain based privacy-preserving system for electric vehicles through local communication," in *Proc. of IEEE Int. Conf. on Communications (ICC),* Ireland, pp. 1–6, 2020.
[46] P. C. Wei, D. Wang, Y. Zhao, S. K. S. Tyagi and N. Kumar, "Blockchain data-based cloud data integrity protection mechanism," *Future Generation Computer Systems,* vol. 102, pp. 902–911, 2020.