Blockchain based Cloud storage framework with Self-Sovereign-Identity and Access Control

Srinivasa Suresh Sikhakolli
Kirloskar Institute of Management,
Pune India

Asha kiran Sikhakolli
Dr. D. Y Patil B-School,
Pune, India

ABSTRACT
The cloud acts as a platform to store more number of data and it is known for its low cost computing and data storing efficiency. In recent time, more number of organizations and individuals utilize cloud storage services to back up the essential information. But, the stored data undergoes different kind of threats related to security issues due to internal and external attacks. Whenever, the data user request access to encrypt the data file, a key to access the file is distributed by the third party. But, when the third party becomes untrusted, the entire security system will be at risk. To overcome this issues, this research proposed a cloud storage framework along with the access control and block chain technique. The proposed research undergoes five stages such as initialization, encryption of data, generating the keys, authentication and authorization and data decryption. Among these stages, the authentication and authorization is performed using the proposed Self- Sovereign-Identity based Access Control (SSIBAC) which verifies the user's authenticity and permits him to access the data. The experimental results show that the proposed method achieved minimum delay of 1.59 s whereas the existing methods such as Block chain based Decentralized Architecture (BDCA) and Block chain-based Multi-authority Access Control (BMAC) achieved delay period of 2.10 s and 1.59 s respectively.

Keywords
Authentication, cloud storage, security, Self- Sovereign- Identity based Access Control.

1. INTRODUCTION
Cloud computing has appeared as a fast growing technology with integration of advances in various fields such as utility computing, distributed computing and service oriented architecture. The cloud computing has created a path to reduce the maintenance cost for customers by chartering the resources instead of purchasing the hardware [1,2]. The cloud customers can access the cloud services to share unlimited resources with the help of servers, storage spaces and other utilities [3]. The centralization of large amount of data in the cloud has the capability to reduce the cost for data management thereby enhancing the flexibility of sharing the data [4]. Moreover, the data stored by the user abandons the control of third parties which effectively minimize the threats caused by internal and external attacks [5]. Storing data as a plain text in cloud storage helps to safeguard the data using access control as a sufficient solution. But, when the third party interferes in the cloud storage environment, it leads to leakage of data and affects security of the stored data. To overcome this issues, the data must be encrypted and stored as a cipher text which has capability to create a huge impact related to security leakage [6,7]. The privacy of the data can be achieved by encrypting the data before externalizing it into the cloud environment. But, this process does not support for searchable encrypted data. Searchable encryption is a type of search method which is processed over the ciphertexts [8]. In searchable encryption, the data is encrypted by the owner and the keywords are extracted from the data then contracted to the cloud environment.

The methods based on searchable encryption is incapable to hold the access control [9]. So, some advancements are processed by focusing on access control which is referred as Attribute-Based Searchable Encryption (ABSE). ABSE is a type of cryptographic algorithm which accomplish the confidentiality among the data along with fine grained access control [10]. The secret key is connected to an attribute specified by the ABSE, which also embeds the access policy in the ciphertext [11]. It can be successfully decrypted once the attribute set corresponds to the access policy. In this system, a user is identified by a number of attributes. The data owner can decide which users can decode the ciphertext by creating an attribute policy, and numerous users who meet the policy's requirements only need to encrypt once [12,13]. There are two types of process involved in ABSE such as direct revocation and indirect revocation. In direct process, user is enabled to particulate the revocation process at the time of encryption whereas in indirect process, the revocation is based on process of updating the keys [14]. Moreover, ABSE algorithm has the capability to support the policies based on decryption to accomplish the revocation process with less complexity and bilinear mappings [15]. This research introduced a new methodology Self-Sovereign Identity (SSI) to preserve the privacy in cloud storage using block chain.

The major contributions of the research are mentioned below:

(1) The cloud storage framework along with the access control and block chain technique is proposed where the accessibility to the user is provided by after the verification process of SSIBAC. The proposed SSIBAC verifies the security without the intrusion of trusted authorities.

(2) The proposed framework utilized Ciphertext-Policy Attribute Based Encryption (CP-ABE) algorithm to verify the secureness of the data and it gives data for generation of keys, access the policy of the user and combine the user details in the block chain structure.

The remaining of the paper is organized in the following way: Section 2 presents the related work of this research and the proposed method is described in Section 3. The Section 4 presents the results and analysis of this research and the conclusion of this research is presented in Section 5.

2. RELATED WORK

Pratima Sharma et al. [16] have introduced a distributed and privacy preserving block chain architecture based on cloud storage which consist of access control and verifying the features related to integrity. The introduced architecture utilized bilinear pairing for the process of generating keys and sets up ciphertext based encryption approach in block chain. The generated keys

utilized the distributed system which does not relied on an individual authority and helps to provide a secured atmosphere. Moreover, the introduced architecture utilized the honeybee optimization algorithm to optimize resource and reduce time period during transmission. However, the introduced architecture does not comprise of revocation process which affects the time reliability of the entire architecture.

Marwan Adnan Darwish et al [17] developed a hybrid algorithm on the basis of block chain to overcome the issues regarding the insufficient privacy. The hybrid technique was used in the process of encrypting data to data centers and a distinctive signature was created at the client side. The generated signature was stored on decentralized set of blocks and the efficiency of the architecture was verified using cloud service infrastructure. The introduced architecture using hybrid algorithm provides reliability and enhance the privacy of the user. However, additional source of power is required to run the introduced framework due to integration of block chain.

Caixia Yang et al. [18] have introduced AuthPrivacyChain which is a block chain based access control framework along with privacy protection. Initially, the identity was created on the basis of address of the node in the block chain. After this, the access control of the generated identity was stored and encrypted in block chain. The user takes responsibility to perform authorization related transactions in block chain and the transaction related authorization was implemented on Enterprise Operation System (EOS). The AuthPrivacyChain was utilized to prohibit the hackers and administrators by protecting the data with authorized privacy. However, the AuthPrivacyChain is incapable to track the leakage of security due to improper maintenance of activity records on block chain entities.

Xuanmei Qin et al [19] have introduced a Block chain based Multi authority Access Control (BMAC) to perform secured sharing of data. BMAC utilized Shamir secret method and permissioned block chain to implement attributed related to multiple authorities. Moreover, the BMAC utilized block chain to ensure secureness among multiple authorities and evaluate token for every individual attributes. The BMAC was capable to record the process involved in access control in a secured manner. However, the issues aroused in communication overhead due to the incapability of BMAC to perform communication with various attribute authorities to gather the attribute keys.

Shangping Wang et al [20] have introduced a secured cloud storage framework with the scheme of access control utilizing Ethereum block chain technology. The introduced scheme is a combined form of Ethereum block chain and CP-ABE. The introduced framework consists of three features such as Ethereum block chain technology where the owner stores the data using smart contracts, second one was based on validating the access of data during the access periods and the third one was based on invocation of smart contract in block chain. The introduced secured cloud storage framework has the capability to access the files at low cost in a secured manner. However, the framework lacks data integrity which was need to be uploaded by the data owner.

3. SELF-SOVEREIGN IDENTITY (SSI) TO PRESERVE THE PRIVACY IN CLOUD STORAGE USING BLOCK CHAIN

This research involves the process of privacy preservation using three blocks such as user, data and verifier. Initially, the data is sent by the user as a verifiable credential and this data gets stored in the cloud space. The stored data from the cloud is verified by the verifier and the data is transmitted based on authenticity of the user. The overall process involved in privacy preservation using block chain is presented in figure 1 as follows:

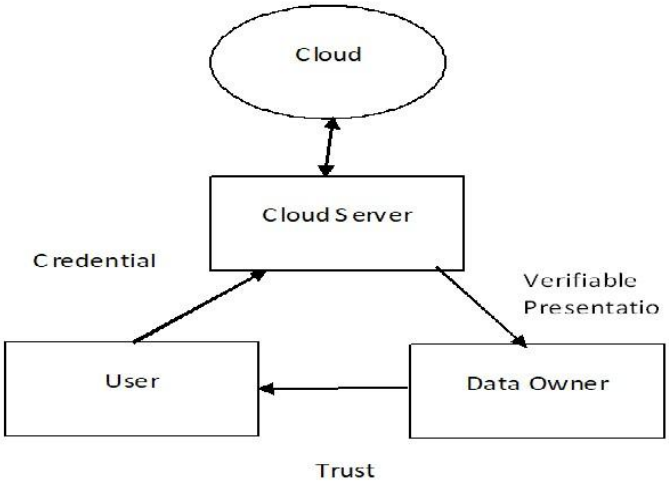

Figure 1. overall process involved in the proposed method

The user has the accessibility to access data when the attributes satisfies access policies of ciphertext. The required file can be decrypted by the user using the secret key to acquire plain text. The data owner (DO) sources data in cloud storage to achieve the dispersal of data files and describes the attribute based policies to encrypt data files of the user. When DO eliminates the unauthorized user, then user lacks his capability to access the data files. Moreover, the DO stores encrypted data files on cloud storage. The cloud storage accesses the stored data files and provides the required services to user through DO.

3.1 Work flow of proposed SSI in preserving the security of cloud data

(i) The attribute authorities and the DO sends request to generate keys and register in the cloud architecture. The key generation function is accomplished using the block chain and creates public key and the master key for DO.

(ii) The request of registration is initiated by the user in the block chain network that sends the information of user to DO and the attribute authorities. The DO and the attribute authorities avail the user's information and create access policies based on the attribute of the user using Cipher text – Attribute based Encryption (CP-ABE) method. The user creates a secret key with the help of DO and the authorized generated keys.

(iii) The encryption is carried out by the DO to encrypt the plain text using access control and create a cipher text. The created cipher text is shared with the authorized attribute during the process of re-encryption. After the process of re-encryption, the authorized cipher text is carried out to the cloud storage space.

(iv) When the user needs to access the data, he sends access request to DO. The DO takes out the required information from the cloud storage space and shares it with the user. Then, user proceeds to decrypt plain text. Whenever the user is

authenticated as an active user, he gets access to decrypt the data as ciphertext.

(v) The authenticated user has the right to alter the attributes based on his requirements and the keys are regenerated to alter the list of user and upgrades the access policies in cloud storage.

3.2 Functions based on smart contract

The proposed architecture sets up different smart contract functions based on CP-ABE algorithm which offers different kind of services such as system initialization, encryption of data files, generation of keys, verification using LSSI and file decryption. The work flow of the fore mentioned process is presented in figure 2 as follows:

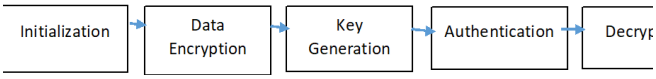

Figure.2 Representation of the work flow

Consider the system with n number of attributes and the bilinear group is represented as $e: G_0 \times G_0 \rightarrow G_T$. Where the bilinear group with prime order is denoted as p and the hash function is represented as H which performs mapping to random attributes of G_0.

3.2.1 Initialization

The DO accomplishes the setup algorithm and outsource public key and the master key which is represented as P_K and M_K of DO. The proposed architecture is based on CP-ABE algorithm which performs attribute based encryption. A bilinear group along with generators and random elements are selected by the DO and distributed as P_K and M_K. The value of P_K and M_K is represented in equation (1) and (2) respectively.

$$P_K = \{G_0, g, g^{\alpha}, g^{\beta}, e(g,g)^{\beta}, H\} \quad (1)$$

$$M_K = \{\alpha, \beta \in Z_p\} \quad (2)$$

3.2.2 Data file encryption

In prior time of uploading the data file in cloud storage, smart contract is positioned by DO to block chain environment. The below mentioned process is employed while uploading the data file the cloud server.

(i) The DO selects inimitable identifier (I_{ID}) for the data file (F). The hash function named Sha-256 is utilized to hash the I_{ID} which is noted as $H(I_{ID})$. The $H(I_{ID})$ gets stored in the block chain by executing the hash function present in smart contract.

(ii) The encryption is performed by the DO in the file (F) using an algorithm known as Attribute Encryption Standard (AES) where the content key C_K is attained in a random manner in specified key space. The encrypted file is recorded as $E_{C_K}(F)$ and upgraded at the cloud storage space.

(iii) The access structure of the encrypted content key E_{C_K} is defined by the DO and E_{C_K} is selected by attribute based encryption algorithm with the access control. The access tree (Π) is selected using attribute based encryption algorithm where leaf nodes are represented as attributes and cipher text C_T is obtained as outcome based on the following algorithm.

$$Encrypt(P_K, C_K, \Pi) \rightarrow C_T$$

The DO choose the polynomial function f_x with degree d_x. In every individual node x in the access tree, the polynomials are selected in a top down way (i.e. from root node R). The threshold is initiated as α_x and β_x for every node in the access tree. The DO selects a random value and sets up $q_r(0) = s$, then the DO selects co-efficient values to get the polynomial value $q_r(x)$. In access tree, the leaf nodes are initiated as X and cipher text of the content key is generated as CT which is represented in equation (3) as follows:

$$CT = \begin{Bmatrix} \Pi, \bar{C} = C_K . e(g,g)^{\beta s}, C = g^s \\ \forall_x \in X: C_x = g^{\alpha q_x(0)}, C_x' = H(att(x))^{q_x(0)} \end{Bmatrix} \quad (3)$$

Where the cipher text is represented as CT and the access tree is denoted as Π. The hash function is denoted as H and the value obtained from the public key is represented as $e(g,g)^{\beta s}$.

It is significant to serialize the cipher text into binary values after creating it. The binary file with ciphertext gets stored in block chain by executing CT function in smart contract. The data user give away the respective CT of various data files by insourcing the valid file hashes by executing the function.

3.2.3 Key generation

The data user makes a request for the DO to access the public key. By accepting the request, the DO allot set of attributes S_a to the data user in a validated access period to the data user in smart contract. The DO implements the key generation algorithm and the generated key is accessible using attribute based algorithm (i.e. $keyGen\ (M_K, S) \rightarrow SK$. The key generation algorithm selects a value (r) in a random manner as which is contained in a set Z_p. Then for every individual attribute, the private key is created which is presented in equation (4) as follows:

$$S_K = \{D = g^{\beta+\alpha r}, \forall j \in S: D_j = g^r . H(j)^{rj}, D_j' = g^{\alpha r_j}\} \quad (4)$$

Where the generated private key is denoted as S_K, the hash function of the attribute j is denoted as $H(j)$ and the randomized value is represented as r.

The DO utilized Diffie-Hellman key exchange protocol to compute the common key on the basis of block chain account. The private key is encrypted in a symmetrical way using AES algorithm with an encryption key. The ciphertext of the encrypted private key is mapped with the respective DO with their address.

3.2.4 Authentication by SSIBAC

The Self- Sovereign- Identity based Access Control (SSIBAC) helps the user to manage the data based on their own attributes and credentials. Those attributes are self-signed by the users and they can represent their data to service providers at the time of requirement. This research utilized SSIBAC as a verifier which verifies the active user by introducing significant ideas and principles related to privacy of the people. The proposed research begins with verifying the public or private entities on the basis of encrypted data on cloud storage. The encrypted data relies on the block chain to get Verifiable Credentials (VC) to the DO. The entities of the block chain gather the VC using the authorized entities from the data user. The decentralized identity of the ABAC is required to be distributed at particular Decentralized Identifiers (DID) using the VCs.

3.2.4.1 System model

The SSIBAC model comprise of a data owner (DO) , cloud storage (S_1), data user (U_1) and a block chain (β). The attributes are provided by the permission validator (P_1) from ABAC. The set of verifiable credentials are represented as VC_1 which is outsourced from the U_1. The attribute of subsets obtained from VC_1 is $A = \{\gamma_1, \dots, \gamma_i\}$. The access function which maps the access control of the DO is represented as (ψ) and it consist of access rules (R) to VP by describing schematic fields from VC_1. The access control policy is more complex because the data consist of sensitive information of the user which helps the third parties to corrupt the essential data of the user. So, the attributes of the user are related to VC that provides specified DID which permits the user to safe guard the data in a private space as VP. These VP has the capability to manage the selective disclosure on the basis of zero knowledge proofs. The essential components of ABAC is combined with SSIBAC and the data owner which relies as policy enforcement point and policy administration point. Moreover, it utilizes the access control policies to permit or reject the resources to reach the storage space. The access control consists of policy decision points which is entrenched to the DO as an external component. The access control of the sensitive data is enabled using the policy interference points and the sensitive data such as user IDs, passwords are stored in a separate cloud space.

3.2.4.2 Authentication process of SSIBAC

The SSIBAC learns the attribute of the user by utilizing the access control decision. The VP maintains zero knowledge proofs. The proposed model attained unlinkability when the data user utilized DID to enhance the security wall of user's data. The SSIBAC authenticates and allows the user to access data by following the below mentioned processes.

(i) The data users with VC are mapped with the permission validators to access the particular required resources using the access control policy.

(ii) When the user is permitted to access the information with minimum access policy, then zero knowledge proof denies the user to access the entire information and prohibits the user for next time.

(iii) The DO generates an account for the obtained VC and the credentials of the user are provided to the block chain to validates the trust value.

The block chain based on identity management in SSI is referred as Hyperledger Indy. The Hyperledger Indy performs communication with the Aries agent to record the DID of each blocks, to describe the format of the issued VC and particulate the VC registration. The Aries agent works on a device which control it or from the data center of the third parties. Moreover, the Aries agent utilized private key of the available entities to present it in a secured way and creates a verified credential to receive the data. The generated VC is sent to the cloud storage and the new connection is made between the Aries agent and the cloud sensor through DID communication. At last, the cloud storage space creates a VP to describe the efficiency of the created VC. The VP created by the secured cloud storage environment performs DID communication and sends it to the Aries agent. The VP is sent to the DO who connects with the Hyperledger Indy to verify if the VP is authorized or not. Thus, the exchange of data is authenticated between the DO and the user in a private and secured way.

3.2.5 Decryption of data

The authenticated data is proceeded with the process of data decryption using the decryption algorithm. At initial stage, the data user gathers the contract address and the ID to upload in the cloud server. The data user gains the permission to access by executing the smart contract function. The data user proceeds the further process when DO sets the valid access time for data. The cipher text is encrypted using the common key for both the DO and the data user to access the data. Since, both of them knows the key to access the data, it must be decrypted. The parameters of CT and SK are decrypted using attribute based algorithm which is mentioned below:

$$Decrypt(P_{K,}\, CT, SK) \rightarrow C_K$$

The decryption is a recursive process which takes place in a down-top manner, so it is significant to describe the process of recursion. The decrypted node is represented as $\text{Decryptnode}(CT, SK, x)$ based on two cases such as x is a leaf node and x is a non-leaf node.

(1) When x is a leaf node, $j = att(x)$. When the individual attribute j is not belongs to the set of attribute S, the $\text{Decryptnode}(CT, SK, x) = \text{Null}$. When the individual attribute j belongs to the set of attribute S, then

$$\text{Decryptnode}(CT, SK, x) = \frac{e\left(g^{r}.H(j)^{rj}, g^{\alpha q_x(0)}\right)}{e(g^{\alpha rj}, H(att(x))q_x(0)} \quad (5)$$

$$= \frac{e(g,g)^{q_x(0).\alpha r} e(H(j),g)^{q_x(0).\alpha r}}{e(g,H(j),g)^{q_x(0).\alpha r}} \quad (6)$$

$$= e(g,g)^{q_x(0).\alpha r} \quad (7)$$

(2) When x is a non-leaf node, the following decryption algorithm is utilized. When, all the nodes of z are the children of F, then $F_z = \text{Decryptnode}\,(\text{CT}, \text{SK}, \text{z})$. When the set of children nodes exist in z, then $F_z \neq null$, if the set does not exists then as F_x will be null. Then, the value of F_x is computed in the following way,

$$F_x = \prod_{z \in S_x} F_z^{\Delta_{j,S'_x(0)}} \quad (8)$$

$$F_x = \prod_{z \in S_x} (e(g,g)^{q_z(0).\alpha r})^{\Delta_{j,S'_x(0)}} \quad (9)$$

$$F_x = \prod_{z \in S_x} (e(g,g)^{q_x(index(z)).\alpha r})^{\Delta_{j,S'_x(0)}} \quad (10)$$

$$F_x = \prod_{z \in S_x} (e(g,g)^{q_x(j).\alpha r})^{\Delta_{j,S'_x(0)}} \quad (11)$$

$$F_x = e(g,g)^{z \in S'_x \sum \alpha r.q_x(j) \Delta_{j,S'_x(0)}} \quad (12)$$

$$F_x = e(g,g)^{\alpha r q_x(0)} \quad (13)$$

Where $j = index\ (z), S'_x = \{index(z) : z \in S_x\}\ and\ \Delta_j S'_x(x) = \prod j \in S'_x$ is known as coefficient of Lagrange interpolation.

After the process of obtaining C_k, the data user decrypt the encrypted data file from cloud server using symmetric encryption algorithm and provides the secured data file based on the requirement of the user.

4. RESULTS AND ANALYSIS

In this section, the performance of the proposed cloud storage architecture using LSSI is analysed. The result is analysed based on two sub sections such as performance analysis and

comparative analysis. In performance analysis, the overall performance of the proposed architecture is evaluated based on the attributes and the run time. In comparative analysis, the efficiency of the proposed method is evaluated based on delay, throughput, computation time and resource utilization.

4.1 Evaluation metrics

The performance of the proposed LSSI is evaluated based on metrics such as delay, throughput, computation time and resource utilization. This section provides description of each parameters utilized in evaluating the performance of the proposed architecture.

(i) Delay

It is defined as the average time period needs to transfer the data to total number of data blocks. The delay is evaluated using the formula presented in equation (14) as follows,

$$D = \frac{\text{Avg.time taken to transfer the data}}{\text{Total number of data blocks}} \quad (14)$$

(ii) Network throughput

It is defined as the ratio of transferred data from one block to another block in a less delay time. The network throughput can be evaluated using the formula represented in equation (15) as follows,

$$N_T = \frac{\text{Total number data block transmitted}}{\text{Minimum time delay}} \quad (15)$$

(iii) Computational time

Computational time is also known as running time which is described as the time needed to perform the overall process.

4.2 Performance analysis

The performance of the proposed method is evaluated based on the time taken to complete one cycle based on the count of attributes which ranges from 2 to 10. The performance of the proposed SSIBAC method is represented in table 1 and the figure 3 presents the graphical representation for evaluation of performance based on time.

Table.1 Performance evaluation table

Parameter	Method	No. of. Attributes				
		2	4	6	8	10
Computational Time (ms)	BDCA [16]	9.32	17.56	24.67	32.54	51.62
	BMAC [19]	8.27	15.21	23.32	30.71	52.67
	SSIBAC	6.58	12.3	21.98	29.56	44.12

The results from table 1 shows that the proposed SSIBAC had took minimum time to transfer the attributes. For instance, the time taken by the existing methods such as BDCA and BMAC ranges from 9.32ms to 51.62ms and 8.27ms to 52.67ms respectively. But, the proposed SSIBAC had took less time period of 6.58ms to 44.12 ms which is relatively lesser than other two techniques. This better result is due to its capability in providing full access to the DO without the help of centralized authority. The graphical representation for the overall performance based on time is shown in figure 3 as follows,

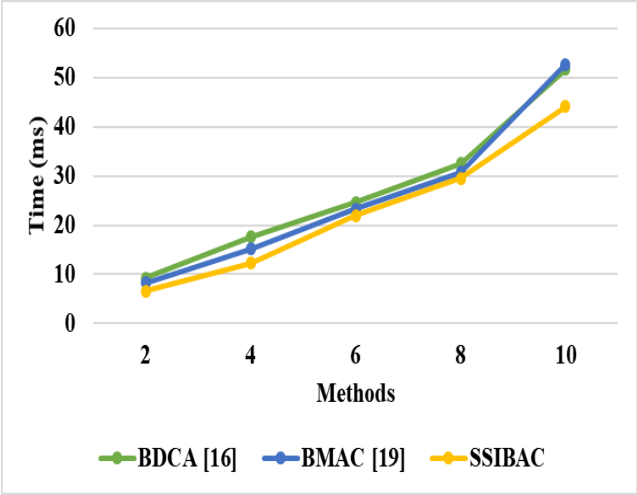

Figure 3. Graphical representation for evaluation of performance based on time

4.3 COMPARATIVE ANALYSIS

In this section, the efficiency of the proposed SSIBAC method is evaluated based on metrics such as delay, throughput, computation time and resource utilization. The description of these parameters are described in section 4.1. The table 2 mentioned below compares the proposed SSIBAC with the existing methods such as BDCA [16] and BMAC [19] based on the fore mentioned evaluation metrics. The overall efficiency of the proposed method is compared with the existing methods for 200 transactions.

Table.2 Comparative table

Methods	Delay (s)	Throughput (Kbps)	Computation time (s)	Resource utilization(%)
BDCA [16]	2.10	4800	18	73
BMAC [19]	1.92	5200	14	71
SSIBAC	**1.59**	**5600**	**12**	**76**

The results from table 2 shows that the proposed method achieved better performance in overall metrics. For example, consider the delay time, the delay time of the proposed SSIBAC is 1.59 s which is comparatively lower than BDCA (2.10 s) and BMAC (1.92 s). The better result of the proposed SSIBAC is due to its efficiency in permitting DO to access the data without any centralized authority.

5. CONCLUSION

The cloud is well known for its cost efficiency which offers massive storage space to help the organizations and individuals to store their data. But, it is essential to consider the privacy and access control before outsourcing the data into cloud space. In order to solve this issue, this research proposed a cloud storage framework along with the access control and block chain technique. The proposed research has five stages in offering accessibility to the user and the five stages include initialization,

encryption of data, generating the keys, authentication and authorization, and data decryption. Among five, authentication and authorization is an essential process to verify the active user. So, this research introduced SSIBAC for the verification process which verifies the user's authenticity and permits him to access the data. The proposed method attains minimum delay time of 1.59 s which is comparatively lesser than the existing methods. The future work will be based on reducing the computational complexities occurred in the proposed method.

6. REFERENCES

[1] Xue, Yingjie, Kaiping Xue, Na Gai, Jianan Hong, David SL Wei, and Peilin Hong. "An attribute-based controlled collaborative access control scheme for public cloud storage." IEEE Transactions on Information Forensics and Security 14, no. 11 (2019): 2927-2942.

[2] Zarezadeh, Maryam, Hamid Mala, and Maede Ashouri-Talouki. "Multi-keyword ranked searchable encryption scheme with access control for cloud storage." Peer-to-Peer Networking and Applications 13 (2020): 207-218.

[3] Ghorbel, Amal, Mahmoud Ghorbel, and Mohamed Jmaiel. "Accountable privacy preserving attribute-based access control for cloud services enforced using blockchain." International Journal of Information Security (2021): 1-20.

[4] Xu, Shengmin, Guomin Yang, Yi Mu, and Ximeng Liu. "A secure IoT cloud storage system with fine-grained access control and decryption key exposure resistance." Future Generation Computer Systems 97 (2019): 284-294.

[5] Saravanan, N., and A. Umamakeswari. "Lattice based access control for protecting user data in cloud environments with hybrid security." Computers & Security 100 (2021): 102074.

[6] Susilo, Willy, Peng Jiang, Jianchang Lai, Fuchun Guo, Guomin Yang, and Robert H. Deng. "Sanitizable access control system for secure cloud storage against malicious data publishers." IEEE Transactions on Dependable and Secure Computing 19, no. 3 (2021): 2138-2148.

[7] Li, Hongzhi, Dezhi Han, and Mingdong Tang. "A privacy-preserving storage scheme for logistics data with assistance of blockchain." IEEE Internet of Things Journal 9, no. 6 (2021): 4704-4720.

[8] Maesa, Damiano Di Francesco, Paolo Mori, and Laura Ricci. "A blockchain based approach for the definition of auditable access control systems." Computers & Security 84 (2019): 93-119.

[9] Varri, Uma Sankararao, Syam Kumar Pasupuleti, and K. V. Kadambari. "Practical verifiable multi-keyword attribute-based searchable signcryption in cloud storage." Journal of Ambient Intelligence and Humanized Computing (2022): 1-13.

[10] Chaudhari, Payal, and Manik Lal Das. "Privacy preserving searchable encryption with fine-grained access control." IEEE Transactions on Cloud Computing 9, no. 2 (2019): 753-762.

[11] Algarni, Sultan, Fathy Eassa, Khalid Almarhabi, Abduallah Almalaise, Emad Albassam, Khalid Alsubhi, and Mohammad Yamin. "Blockchain-based secured access control in an IoT system." Applied Sciences 11, no. 4 (2021): 1772.

[12] Qin, Xuanmei, Yongfeng Huang, Zhen Yang, and Xing Li. "LBAC: A lightweight blockchain-based access control scheme for the internet of things." Information Sciences 554 (2021): 222-235.

[13] Kesarwani, Abhishek, and Pabitra Mohan Khilar. "Development of trust based access control models using fuzzy logic in cloud computing." Journal of King Saud University-Computer and Information Sciences 34, no. 5 (2022): 1958-1967.

[14] Ren, Yongjun, Fujian Zhu, Jian Qi, Jin Wang, and Arun Kumar Sangaiah. "Identity management and access control based on blockchain under edge computing for the industrial internet of things." Applied Sciences 9, no. 10 (2019): 2058.

[15] Deep, Gaurav, Rajni Mohana, Anand Nayyar, P. Sanjeevikumar, and Eklas Hossain. "Authentication protocol for cloud databases using blockchain mechanism." Sensors 19, no. 20 (2019): 4444.

[16] Sharma, Pratima, Rajni Jindal, and Malaya Dutta Borah. "Blockchain-based decentralized architecture for cloud storage system." Journal of Information Security and Applications 62 (2021): 102970.

[17] Darwish, Marwan Adnan, Eiad Yafi, Mohammed A. Al Ghamdi, and Abdullah Almasri. "Decentralizing privacy implementation at cloud storage using blockchain-based hybrid algorithm." Arabian Journal for Science and Engineering 45 (2020): 3369-3378.

[18] Yang, Caixia, Liang Tan, Na Shi, Bolei Xu, Yang Cao, and Keping Yu. "AuthPrivacyChain: A blockchain-based access control framework with privacy protection in cloud." IEEE Access 8 (2020): 70604-70615.

[19] Qin, Xuanmei, Yongfeng Huang, Zhen Yang, and Xing Li. "A blockchain-based access control scheme with multiple attribute authorities for secure cloud data sharing." Journal of Systems Architecture 112 (2021): 101854.

[20] Wang, Shangping, Xu Wang, and Yaling Zhang. "A secure cloud storage framework with access control based on blockchain." IEEE access 7 (2019): 112713-112725.