Survey of Centralized and Decentralized Access Control Models in Cloud Computing

Suzan Almutairi[1], Nusaybah Alghanmi[2], Muhammad Mostafa Monowar[3]
Technical and Vocational Corporation, Saudi Arabia[1]
Department of Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia[2, 3]

***Abstract*—In recent years, cloud computing has become a popular option for a number of different businesses sectors. It is a paradigm employed to deliver a range of computing services, such as sharing resources via the Internet. Security issues in cloud computing necessitates the need for a mechanism to keep the system safe and reliable. An access control mechanism is one that permits or denies access to cloud services. This paper presents a survey of access control models in Cloud Computing. Several existing surveys on access control mechanisms in cloud computing mainly focused on traditional access control models and encryption-based access control models while the others focused on applying blockchain technology in cloud access control. However, access models possess different characteristics, such as the system's reliance on a centralized cloud trusted system administrator to manage the access policy or adopting decentralized approach. This paper reviews and analyses existing access control mechanisms in cloud computing, based on centralized and decentralized access control models, provides detailed comparisons on each model's advantages and limitations, and discusses the challenges of, and future research direction for access control.**

Keywords*—Cloud computing; access control; cloud security; centralized; decentralized

I. Introduction

Cloud computing has recently become the information technology (IT) foundation for many companies and organizations due to many its benefits, such as its interoperability, mobility, and cost effectiveness. Cloud computing technology refers to the virtualization of the IT infrastructure, including the hardware, software, and networking. This IT infrastructure is associated and designed together to provide the cloud services to the end user via the Internet [1-3]. Cloud computing includes three service models:

- Software as a service (SAAS): This model offers a variety of applications to the user;
- Platform as a service (PAAS): The model offers infrastructure as an environment for the developer to develop their own services or applications;
- Infrastructure as a service (IAAS): The model offers a virtualized resource, such as database services, a virtual machine, and a storage service to the user.

Securing the cloud environment is a critical issue, due to the unique characteristics of cloud computing, such as multitenancy and elasticity of the sharing of resources. It requires an access mechanism to ensure confidentiality, integrity, and availability for cloud data [4, 5]. Access control is considered to be the first line of defense of the system, and allows authorized users to gain access to the protected information and system resources, as well as denying unauthorized users access to the same.

In cloud computing, access control permits cloud users to access specific applications, or to protect data privacy, and can also be applied to protect the cloud user's resources. Finally, it gives the correct access permissions to each level of service. For example, in IAAS, it separates access permission to the guest's virtual machine from the host operating system. Access control in the cloud can be formed as either centralized or decentralized access control models. The former depends on a central authority to manage the access policies and key generation, while the latter depends on a multi-authority to manage the keys, and to store encrypted resource and access policies.

A. Contribution

Several recent works presented survey papers concerning the traditional access control models and encryption-based access control models for cloud computing, [1, 6]. Also, Xie et al. [7] focused on blockchain technology in the cloud, and its associated issues. However, no extant work presented a survey from the way the access control policies are managed (centralized and decentralized). The contributions of this paper are therefore as follows:

- We provide a taxonomy, based on centralized and decentralized access control models;
- We present detailed comparisons of the existing solutions of centralized and decentralized access control models, and the strengths and the limitations of each;
- We discuss the challenging issues with the existing access control models, and provide future directions.

B. Motivation

The proposed access control models usually possess two main characteristics [8]: first, they require one or more centralized centers to store or manage different data, such as user identities, cryptographic keys, and access rights etc. Second, all three cloud service models require a cloud trusted system administrator to manage the access rights and authorization process for other users. Consequently, there are two issues: first, an attack on the centralized center, causing single point of failure resulting in data compromise. In this case, the attacker may tamper with the data access, steal the

resources, or cause other forms of damage. Second, a malicious cloud security administrator could use their authority to access resources illegally, or to tamper with legal users' access rights, which would engender a loss of confidence and trust in the cloud.

In order to reduce the effects of these issues, the researchers start to look at new techniques to decentralize the cloud storage of access control. The blockchain or multiple distributed authority are an example of these techniques. The blockchain technique is more difficult to manage compared to centralized access control. Despite that, they are more secure in key distribution, file information, etc. Another issue is blockchain is slower than centralized access control because of the blockchain design nature [9].

This study therefore reviews and analyses the relevant literature on existing access control mechanisms in cloud computing that concern centralized and decentralized access control models, and assesses their advantages and disadvantages.

The rest of this paper is organized as follows: Section II introduces the background to access control models, blockchain technology, and smart contract techniques. Section III presents the existing solutions for access control in centralized and decentralized models in cloud computing, while Section IV discusses the challenges of, and potential future research direction for access control models in cloud computing. Finally, Section V concludes the paper.

II. Background

This section provides the background to this paper, including the concept of access control, blockchain technology, and smart contracts.

A. Access Control

This subsection presents the basic elements of access control, access control models, and encryption based access control.

1) Basic elements: There are three elements involved in the access control model, namely subject, object, and access rights [10]. The subject is the entity (users or applications) that can access an object, while the object is a resource, such as files or directories, that requires access. Lastly, access rights include the access policy, such as read or write, from the subject to the object.

2) Traditional access control models: Access control models are described as either discretionary or non-discretionary, and there are three main types, namely discretionary access control (DAC), role-based access control (RBAC), and mandatory access control (MAC) [11]. In DAC, the object's owner is required to specify the subject and the associated privileges and access policy. In RBAC, the access policy is based on the role of the subject. In MAC, the subject sensitivity label is compared with the object sensitivity label, and the former must be equal to or higher than the latter.

3) Encryption based access control: Cryptography algorithms are used to store and protect data as ciphertext, in order that the data remains secure in the cloud. Thus, encryption-based access control assists with achieving complementarity by combining cryptography algorithms with policy-based access control [6]. Attribute-based encryption (ABE) is a leading model of encryption-based access control.

In ABE, the identity of the user is defined by a set of attributes, and is categorized into key-policy ABE (KP-ABE) and ciphertext-policy ABE (CP-ABE) [6]. In a CP-ABE system, a set of attributes are assigned to the user's private key and access structures are bounded in ciphertext. For the decryption of the ciphertext, the user's attributes are matched with the access structure. Meanwhile, in KP-ABE, in the phase of generating the key, the access structure is associated with the user's private key.

B. Blockchain Technology and the Smart Contracts Technique

This subsection presents the basic characteristics of blockchain technology and the smart contracts technique. A blockchain is a sequence of connected blocks. In basic terms, the block is a data container that holds multiple fields within it, such as data transaction, the hash value of the current block, the hash value of the previous block, and the timestamp [12]. It is like a database that holds every transaction as a record. Furthermore, the blockchain has multiple characteristics [7] such as:

- Decentralized: The blockchain is a distributed network, which means that everyone who has been authenticated in the blockchain can participate, and can maintain the entire blockchain;
- Immutable: It is theoretically impossible to change or edit a transaction in the blockchain. This is because of the consensus mechanism, in which every block needs to compute a cryptography puzzle within 10 minutes to be added as a new chain. This new block is broadcast to the network, and other participants verify the correctness of the new block and the transactions it contains. After the block has been verified, it is added to the chain. In addition, a new hash value is generated for the current block, and a hash value for the previous record. The operations are on-going. As a result, any tampering with any block hash value will have to change all the subsequent blocks within a specific timeframe. Therefore, any tampering in any block can be detected immediately;
- Anonymity of user identity: The block in the chain has a unique wallet address. These addresses are generated using public and private pair keys. Every transaction occurs using the user wallet address.

In the same context, the smart contracts technique is a type of computer program that can be auto-generated, based on author codes, and cannot be modified once is generated and deployed, without the need for human interaction. It is used in the blockchain to store the encrypted contract that contains the keyword index (hash value), and other related data. The encrypted keyword, which is a unique hash identifier, helps in searching the service quickly and in retrieving only the correct

results. Consequently, the fee of the service is calculated and deducted from the customer contract. This technique solves the problem in centralized cloud access control of the retrieval of incorrect results, or no results at all, to save cloud computational resource costs [13].

III. Access Control Models in Cloud Computing

This section presents the current centralized and decentralized access control solutions in cloud computing. The former require a central authority to manage the policy and the keys, and to store the data, while the latter require a distributed authority.

A. Centralized Access Control

In the centralized access control model, there are four entities: the data owner (DO), central authority (CA), cloud server (CS), and data user (DU), as shown in Fig. 1. The CA is a trusted center, responsible for the centralized concept, which manages the access policy and generates the keys for the DU and the DO. The DO uploads the relevant resources and stores the data to the CS, while the CS stores the data and provides a transmission service between the DU and the DO. Finally, the DU receives and downloads the required data, according to the access policy.

Liu et al. [14] suggest an online/offline CP-ABE scheme in mobile cloud computing for improving the computational overheads of E-Healthcare Records (EHR). Offline encryption allows major computation, and ensures that the computation online encryption task is reduced. In the scheme proposed, the data encryption is formed of two phases, namely online and offline encryption. In the offline encryption, the EHR owner (Internet of Things (IoT) device) encrypts the data so that intermediate ciphertext is produced, which is subsequently used in the online encryption. In the online encryption, once the data is ready, the access policy is specified by the owner, and the final ciphertext is sent to the cloud. The scheme proposed is superior to other schemes in regard to the online encryption and decryption costs. However, the trade-off between the computation time and storage space should be addressed.

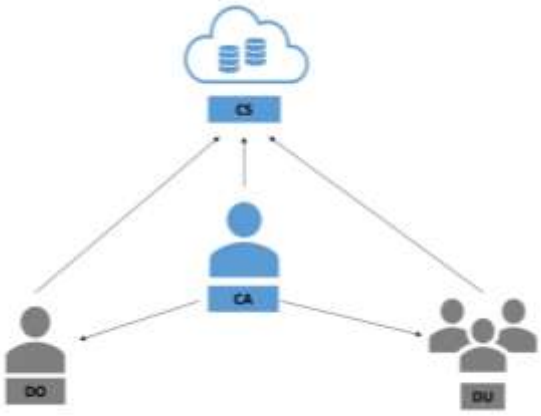

Fig. 1. Centralized Access Control Model

Meanwhile, Lin et al. [15] propose the use of PriGuarder to protect data privacy in the cloud. The approach proposed has three stages: DU registration, data creation, and DU access. In each stage, there are two access modes: direct, or anonymous. In the direct access mode, the operation occurs between the DU and the CS. In the anonymous access mode, the operation occurs between the DU and a trusted third party (TTP), with the TTP converting the DU data, identity, or access policy using an attribute fuzzy grouping (AFG), and sends the converted result to the CS. In the DU registration, the DU chooses the access mode to generate the DU identity. Then, in the data creation stage, the DO transmits their data, along with a statement of policy rights, and the chosen access mode. Finally, the verification of the DU occurs in the data access stage, according to the access mode selected. The key power in PriGuarder is the AFG method, which protected user privacy. However, the study fails to consider that a possible malicious TTP might contravene the users' privacy.

Generally, the hierarchy CP-ABE classifies the related attributes into different attribute trees. In their study, Li et al. [16] provide an efficient extended file hierarchy CP-ABE scheme (EFH-CP-ABE), in which they overcome the issue of encrypting multiple files with a similar access level. As shown in Fig. 2 [16], the EFH-CP-ABE scheme has four entities, namely the DO, the CA, the CS, and the DU. The CA generates three keys: a secret master key, a public key, and a private key for each user. The CS provides the transmission service and stores the ciphertext in the storage, while the DO stores and shares the data with the CS. Finally, the DO entity divides the data (m) into different blocks, such as m_1, m_2, and mi, and provides different session keys (sk) such as sk_1, sk_2, and sk_i. The DO randomly encrypts (E) the block mi with ski to produce $E_{sk}(M)= \{E_{sk1}(m_1), E_{sk2}(m_2), E_{sk3}(m_3)\}$, which is then stored in the CS. Finally, the DU downloads the ciphertext and decrypts part or all of the ciphertext, according to the node level's particular attributes. The results of the study demonstrate that the access control scheme achieved security and flexibility for cloud storage users. However, the computation time required for the encryption and decryption in the scheme requires improvement.

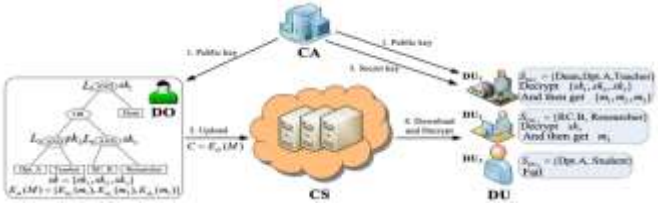

Fig. 2. EFH-CP-ABE Architecture [16].

Meanwhile, Jamal et al. [17] suggest a solution that provides a backup authority node in case of failure, and an efficient method of data access. This is an agent-based ABE access control method that employs the encrypted policy ABE mechanism, and has seven entities. First, the certification authority functions as a certificate issuing agent. Second, there are multiple attribute authorities, each with responsibility for delivering the encryption and decryption keys to the DO and the certified DUs. Third, the client site agent passes the user requested data to the server site agent to retrieve their data

from the cloud storage. Fourth, the server site agent requests are processed at the server-site. Fifth, the client storage server provides storage services and computational resources to the DU and DO. Sixth, the authorized agent keeps track of the neighbour nodes; should there be a certificate authority failure, the authority's back node is activated, subject to a request. Finally, the request handler has responsibility for the data access processing, using shared cache memory scheduling. The approach proposed is secure against malware injection, identity theft, collusion, and certification authority failure attacks. Additionally, the reading response times when accessing the cloud data are improved. However, the process of selecting the appropriate backup authority node needed to be secure.

In their work, Anilkumar and Subramanian [18] propose an algorithm called PB-FGAC, which combines a predicate-based access control (PBAC), and a fine-grained based access control to Swift object storage of the OpenStack cloud platform. PB-FGAC provides fine-grained access control to a predicate, which is part of the object, and helps to avoid access to the whole object. The user requested the OpenStack cloud receives by the NOVA component (NOVA is responsible to compute instances) to process the request, and the request is then forwarded to the object attribute storage services and a policy engine service, respectively. Each policy consists of object-level access and user-level access. After this, the PBAC service helps the user to access the specific data or predicate required. The results of the study demonstrate that the model provides more restricted access control than other environments with a default access control policy, such as the Amazon Web Services (AWS), Microsoft Azure, and Open Stack cloud platforms. However, the model proposed only applies to the JavaScript Object Notation (JSON) document, and its confidentiality also required addressing.

Finally, Ghaffar et al. [19] discuss the security issues associated with the data management operations in centralized cloud storage, such as insider attack, the lack of a data access verification model, and the lack of a sharing data configuration. They propose a modified model for data access and sharing in cloud storage, using a proxy key protocol. This model involves three entities, the DO, the DU, and the CS, together with three phases, namely data access, data storage, and a data sharing system. The data access involves the user and cloud server's authentication mechanism, and these then communicates with each other by sharing the session key. The data storage provides the users with the storage services required to share encrypted files or data with other desirable users. The data-sharing system searches by the user's keywords for the encrypted file after the user is authenticated. The method proposed is secure against multiple attacks, such as user or cloud impersonation and man-in-the-middle attacks, and data confidentiality breaches. However, the DU could search for a specific file by entering related keywords, resulting in the retrieval of multiple files, or a long search time.

Table I compares the centralized access control models, based on different criteria. The data confidentiality ensures the data is not a disclosure by unauthorized users. The scalability ensures when the number of users is increased, the system performs well.

B. Decentralized Access Control

In the decentralized access control model, there are four entities: the DO, the distributed center, the CS, and the DU, as shown in Fig. 3. A distributed center is a trusted authorization database that manages the access policy, and generates the keys for the cloud, the DU, and the DO. The distributed center can be a blockchain, a distributed CA, or a distributed attribute authority (AA). The DO uploads resources and stores the data to the CS, and publishes the access rights to the distributed server. Meanwhile, the CS stores the data, provides the authentication, and determines the permission of the DU and the DO. Finally, the DU receives and downloads the required data or resource, according to the access policy.

Al-Dahhan et al. [20] propose a distributed multi-authority CP-ABE. The system propose contains six entities and three phases. It commenced with an initialization phase, in which the CA must register each authorized DU and AA to obtain their identities. The DO creates an encrypted access control policy for the DU. Then, the DO publishes the encrypted access policy to the CS. One of the distributed AAs in the system then generates the keys for the DU to decrypt the access policy. The DU could then access and decrypt the resource, if they are authorized and matched the attribute assigned to him. If the user revokes his authorization, the CA informs the proxy server responsible for maintaining and updating the DU authorization list. The system proposed protects data confidentiality and secured against collusion attack. However, the DU performs a lot of computation for computing the secret key from one of the distributed AA.

TABLE I. A COMPARISON OF THE EXISTING MODELS FOR CENTRALIZED ACCESS CONTROL IN CLOUD COMPUTING

The proposed model	**Access Control**		**Data confidentiality**	**Scalability**
	Traditional Access Control	***Encryption- based***		
Online/offline CP-ABE scheme [14]	X	✓	✓	X
PriGuarder [15]	X	✓	✓	X
EFH-CP-ABE [16]	X	✓	✓	X
Agent-based ABE access control [17]	X	✓	✓	X
PB-FGAC [18]	✓	X	X	X
Ghaffar et al. [19]	X	✓	✓	X

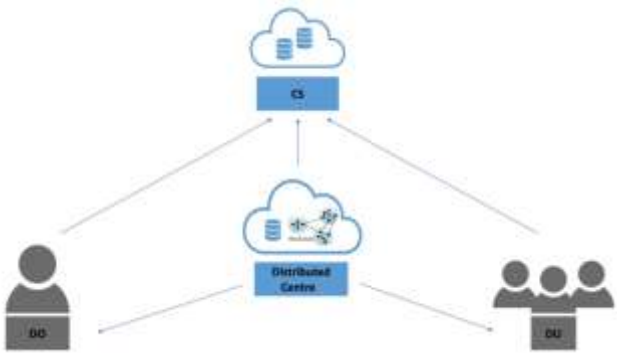

Fig. 3. Decentralized Access Control Model.

In another study, Wei et al. [21] propose a multiauthority CP-ABE to protect the outsourced data in the cloud. The approach proposed consists of five entities, each of which had a role. First, the third party produces a global parameter for the approach proposed. Second, each AA is responsible for generating a public parameter and the secret master key, and for managing the validity of each user's attribute belong to its area. Third, the DO defines its access policy on the DU attribute, and sends the encrypted data to the cloud. Fourth, every DU has a unique global identifier, set of attributes, and secret key. Finally, the CS is responsible for storing the data and updating any information requested from any entity. The proposed approach is secure against collusion attacks. However, the number of attributes has an influence on the efficiency of the scheme's algorithm.

Wang et al. [22] propose an integrated technique that combined blockchain, smart contracts, and CP-ABE. The model proposed consists of four entities, the DO, the DU, the CS, and blockchain, and involved four phases, namely system initialization, file encryption, key generation, and file decryption. The DO is responsible for defining the access control policies, determining the attribute sets, and creating the smart contract with valid access periods to the DU. Meanwhile, the DU is required to satisfy the encrypted smart contract's attribute set to decrypt it, then to obtain the content key to decrypt the stored encrypted files, and access them in the cloud. The CS is responsible for storing the encrypted resources or files, and finally blockchain is responsible for deploying and storing the smart contracts. The cost of the data access is low, and the performance is feasible. However, the study does not consider the integrity of the file uploaded by the DO.

In another study, Yang et al. [8] propose a mechanism named Authprivacychain, which uses the blockchain technique, merged with smart contracts. The model proposed consists of four entities, the DO, the DU, the CS, and blockchain., It involves four phases, namely initialization, access control, authorization, and authorization revocation. First, the DO uploads the resources required to the cloud and registered the transaction in the blockchain, then publishes the authorization to the DU. The DU then sends a request to the CS for a specific resource, and the CS sends a query to the blockchain, and evaluates whether the resource requested by the DU has permission or not. Finally, the CS replies to the request with the stored access. These processes are designed in an encrypted manner to secure data privacy. The approach proposed is secure against external and internal attacks, and the Authprivacychain protects the confidentiality, integrity, and accountability of the resources, as well as the availability and authenticity. However, the time performance of the technique proposed depends on the blockchain node configuration. This is because there are different types of the blockchain with various configuration parameters. For example, block time generation, a block take two minutes while other take 10 mintues [9].

The difference between [8] and [22] is in the decryption of the resource. The DU could access the resources once they are authorized by the DO, and the decryption key is delivered at the initialization phase in [8], but in [22] the DU is required to satisfy the attribute set by the DO, in order to obtain the decryption key of the encrypted resource.

Meanwhile, the authors of [23] propose a decentralized attribute ABE access control model for a mobile cloud. The scheme proposed is similar to that in [18], but the algorithm specification differed. The proposed approach is secured against reply and collusion attacks. However, it fails to achieve a public ciphertext test. Furthermore, the computational complexity increases with the attribute number involved in the ciphertext.

Table II compares the decentralized access control models, including the different key features used in the access control models, namely the access control permission, the authorization, the authorization revocation, the authentication identity, and the access log. In terms of the access control permission, it specifies what right the user has to the data, such as read or write. Meanwhile, in terms of authorization, it specifies the permission required for a user to access the cloud resource or data. The authorization revocation refers to the authorized users' ability to revoke, dispense, or delete the resource access permission of other permitted users. The authentication is also the mechanism used to verify that the user is who they claim to be. Finally, the access log contains all the information of all the requests for user actions on the cloud resource. Table III summarizes the advantages, limitations, and type of architecture of the existing access control models in cloud computing.

TABLE II. A COMPARISON OF THE EXISTING MODELS FOR DECENTRALIZED ACCESS CONTROL IN CLOUD COMPUTING

The proposed model	Techniques	Access control permission	Authorization	Authorization revocation	Authentication identity	Access log
Al-Dahhan et al. [20]	Multi-authority CP-ABE	✓	✓	✓	A symmetric key	X
Wei et al. [21]	Multi- authorityABE	✓	✓	✓	Global identifier	X
Wang et al. [22]	blockchain , smart contract , CP-ABE	✓	X	X	blockchain wallet address	✓
Authprivacychain [8]	blockchain , smart contract	✓	✓	✓	blockchain wallet address	✓
De et al. [23]	Multi- authorityABE	✓	✓	✓	Global identifier	X

TABLE III. A COMPARISON OF ACCESS CONTROL MODELS IN CLOUD COMPUTING

Year	Access Control Model	Architecture	Advantages	Limitations
2018	Online/offline CP-ABE [14]	Centralized	Improved the computational overheads of the current schemes for EHR.	Trade-off between the computation time and storage space required addressing.
	PriGuarder [15]	Centralized	The key power in PriGuarder was the AFG method, which protects user privacy.	A malicious TTP may disclosure users' privacy.
	Al-Dahhan et al. [20]	Decentralized	Protected data confidentiality and secured against collusion attacks.	The computation cost for computing the secret key generation from the AA was the responsibility of the DU.
	Wei et al. [21]	Decentralized	Secure against collusion attacks.	The number of attributes had an influence on the efficiency of the scheme's algorithm.
2019	EFH-CP-ABE [16]	Centralized	Encrypted multiple files at a similar access level.	Computation time for the encryption and decryption required enhancement.
	Agent-based ABE access control [17]	Centralized	Secure against various attacks, such as malware injection, identity theft, collusion, and certification authority failure attacks. The response time for reading the data access in the cloud was improved.	The process of selecting the appropriate backup authority node required securing.
	Wang et al. [22]	Decentralized	Low cost of the data access, and feasible performance.	Did not consider the integrity of the file uploaded by the data owner.
2020	PB-FGAC [18]	Centralized	Provided partial access to the JSON document.	Only applied to the JSON document. Confidentiality also required addressing.
	Ghaffar et al. [19]	Centralized	Secure against multiple attacks, such as user or cloud impersonation, and man-in-the-middle attacks. Provided data confidentiality.	Searching by DU for a specific file resulted in the retrieval of multiple files, or wasting time.
	Authprivacychain [8]	Decentralized	Secure against internal and external attacks. Protected the confidentiality, integrity, and accountability of the resources, as well as availability and authenticity.	The performance depended on the blockchain node configuration.
	De et al. [23]	Decentralized	Secure against reply and collusion attacks.	Failed to achieve the public ciphertext test. The computational complexity increased with the attribute number involved in the ciphertext.

IV. CHALLENGES AND FUTURE RESEARCH DIRECTION FOR ACCESS CONTROL MODELS IN CLOUD COMPUTING

Section III discussed the different solutions currently proposed for access control models in cloud computing. However, certain issues and limitations remain that must be addressed by future studies. This section presents the challenges and suggested future research direction for both centralized and decentralized access control models in cloud computing.

A. Centralized Access Control

Section III-A presented the current solutions based on centralized access control, but the studies discussed relied on either traditional access policy models or encryption based models, both of which are insufficient for securing a system. More hybrid works that combine both access policy models and encryption models are required to ensure confidentiality and integrity.

Importantly, the studies in centralized access control assume that a CA or an AA is trusted. While, the CA or AA is not always trusted, which causes a number of issues, such as leakage of sensitive data and keys. Thus, it leads to disclosure users' privacy. Also, the CA may suffer from failure for any reasons such as attacks but the current process [17] of selecting the appropriate backup authority node required securing.

Moreover, in terms of centralized cloud systems, the extant studies employed multiple techniques and algorithms, such as public-key encryption, symmetric encryption, and hash algorithms. These studies store the resources in an encrypted manner in the cloud storage, however, searching a particular resource or a specific file by entering related keywords [19], resulting in the retrieval of multiple files, may produce erroneous outcome, or an extended search time.

Furthermore, the computation time for the encryption and decryption is a concern which needs to be enhanced when encrypting multiple files at a similar access level. However, in mobile cloud computing, trade-off between the computation time for the encryption and decryption and storage space required addressing.

Finally, the number of users in the cloud system is increased day by day; thus more attention is needed for scalability to ensure the system's reliability.

B. Decentralized Access Control

Currently, the concept of adopting distributed authority to tackle the issue of centralized authority is ongoing, such as the work of [8, 22, 23]. However, limitations remain that must be addressed, such as the restoration of data from a compromised AA or CA.

The blockchain (decentralized) uses a hash function to generate a unique id for each resource and to store it in a smart contract, which means that the unique id provides a quick search with privacy, and returns the correct result [13]. Therefore, the reading requests in decentralized cloud models are faster than in centralized cloud models. There is a need for more focus on the application and development of blockchain technology and smart contracts to overcome their existing limitations, such as the blockchain node configuration, slow performance in writing access rights, blockchain node registration, and computational resources.

Furthermore, the authorization revocation discussed in Section III-B requires different reasons to perform, such as a user sending a request for the revocation. Also, the resource may be compromised, and the cloud server may therefore wish to disable it. The process of revocation therefore requires more attention to ensure that it is secure and fast [8].

Finally, the access log contains records of all the entities' interactions in the system. It is of essential value, as it can provide a mechanism to locate the interaction responsible for an attack on the system. Moreover, security analysis can perform post-audit analysis on these records to detect any vulnerabilities in the system. Consequently, there is a need to involve access log design within the system implementation [8].

V. Conclusion

Cloud computing provides different services via the Internet that may engender a number of security issues, such as unauthorized access to cloud resources. Access control is a security technique that controls the access to cloud services. This paper provided a taxonomy for access control models in cloud computing, according to centralized and decentralized models, and discussed a range of works that employed this taxonomy, comparing the advantages and limitations of each.

The study concluded that, in order to improve the security of access control models in cloud computing, there is a need for different solutions, both centralized and decentralized. In centralized access models, the system should ensure both confidentiality and integrity, and tackle the issue of long search time. While in decentralized access models, authorization revocation must be faster and more secure. Moreover, the use of access logs to keep track of any disclosure of the system is required, and blockchain node configuration must be enhanced to improve the system's performance. The study also presented the current challenges and recommended a direction for future research in access control models in cloud computing.

References

[1] K. Albulayhi, A. Abuhussein, F. Alsubaei, and F. T. Sheldon, "Fine-Grained Access Control in the Era of Cloud Computing: An Analytical Review," in 2020 10th Annual Computing and Communication Workshop and Conference (CCWC), 2020, pp. 0748-0755.

[2] V. Winkler, "Chapter 2 - Cloud Computing Architecture," in Securing the Cloud, V. Winkler, Ed. Boston: Syngress, 2011, pp. 29-53.

[3] V. Winkler, "Chapter 1 - Introduction to Cloud Computing and Security," in Securing the Cloud, V. Winkler, Ed. Boston: Syngress, 2011, pp. 1-27.

[4] P. J. Kaur and S. Kaushal, "Security concerns in cloud computing," in International Conference on High Performance Architecture and Grid Computing, 2011, pp. 103-112: Springer.

[5] S. Curry et al., "Infrastructure security: Getting to the bottom of compliance in the cloud.," The Security Division of EMC (2010).

[6] F. Cai, N. Zhu, J. He, P. Mu, W. Li, and Y. Yu, "Survey of access control models and technologies for cloud computing," Cluster Computing, vol. 22, no. 3, pp. 6111-6122, 2019/05/01 2019.

[7] S. Xie, Z. Zheng, W. Chen, J. Wu, H.-N. Dai, and M. Imran, "Blockchain for cloud exchange: A survey," Computers & Electrical Engineering, vol. 81, p. 106526, 2020/01/01/ 2020.

[8] C. Yang, L. Tan, N. Shi, B. Xu, Y. Cao, and K. Yu, "AuthPrivacyChain: A Blockchain-Based Access Control Framework With Privacy Protection in Cloud," IEEE Access, vol. 8, pp. 70604-70615, 2020.

[9] M. Schäffer, M. Di Angelo, and G. Salzer, "Performance and Scalability of Private Ethereum Blockchains," 2019, pp. 103-118.

[10] W. Stallings, L. Brown, M. D. Bauer, and A. K. Bhattacharjee, "Chapter 4 - Access Control " in Computer security: principles and practice: Pearson Education Upper Saddle River, NJ, USA, 2012, pp. 113-154.

[11] V. Winkler, "Chapter 5 - Secure the Cloud: Data Security," in Securing the Cloud, V. Winkler, Ed. Boston: Syngress, 2011, pp. 125-151.

[12] S. Pavithra, S. Ramya, and S. Prathibha, "A Survey On Cloud Security Issues And Blockchain," in 2019 3rd International Conference on Computing and Communications Technologies (ICCCT), 2019, pp. 136-140.

[13] S. Wang, Y. Zhang, and Y. Zhang, "A Blockchain-Based Framework for Data Sharing With Fine-Grained Access Control in Decentralized Storage Systems," IEEE Access, vol. 6, pp. 38437-38450, 2018.

[14] Y. Liu, Y. Zhang, J. Ling, and Z. Liu, "Secure and fine-grained access control on e-healthcare records in mobile cloud computing," Future Generation Computer Systems, vol. 78, pp. 1020-1026, 2018/01/01/ 2018.

[15] L. Lin, T. Liu, S. Li, C. M. S. Magurawalage, and S. Tu, "PriGuarder: A Privacy-Aware Access Control Approach Based on Attribute Fuzzy Grouping in Cloud Environments," IEEE Access, vol. 6, pp. 1882-1893, 2018.

[16] J. Li, N. Chen, and Y. Zhang, "Extended File Hierarchy Access Control Scheme with Attribute Based Encryption in Cloud Computing," IEEE Transactions on Emerging Topics in Computing, pp. 1-1, 2019.

[17] F. Jamal, M. T. Abdullah, Z. M. Hanapi, and A. Abdullah, "Reliable Access Control for Mobile Cloud Computing (MCC) With Cache-Aware Scheduling," IEEE Access, vol. 7, pp. 165155-165165, 2019.

[18] C. Anilkumar and S. Subramanian, "A novel predicate based access control scheme for cloud environment using open stack swift storage," Peer-to-Peer Networking and Applications, 2020/07/26 2020.

[19] Z. Ghaffar, S. Ahmed, K. Mahmood, S. H. Islam, M. M. Hassan, and G. Fortino, "An Improved Authentication Scheme for Remote Data Access and Sharing Over Cloud Storage in Cyber-Physical-Social-Systems," IEEE Access, vol. 8, pp. 47144-47160, 2020.

[20] R. R. Al-Dahhan, Q. Shi, G. M. Lee, and K. Kifayat, "Revocable, Decentralized Multi-Authority Access Control System," in 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), 2018, pp. 220-225.

[21] J. Wei, W. Liu, and X. Hu, "Secure and Efficient Attribute-Based Access Control for Multiauthority Cloud Storage," IEEE Systems Journal, vol. 12, no. 2, pp. 1731-1742, 2018.

[22] 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.

[23] S. J. De and S. Ruj, "Efficient Decentralized Attribute Based Access Control for Mobile Clouds," IEEE Transactions on Cloud Computing, vol. 8, no. 1, pp. 124-137, 2020.