Using Attribute-Based Access Control, Efficient Data Access in the Cloud with Authorized Search

Original Scientific Paper

K. S. Saraswathy
Manonmaniam Sundaranar university,
Department of Computer Science,
Abishekapatti, Tirunelveli – 627012.
789saraswathy@gmail.com

S. S. Sujatha
Manonmaniam Sundarnar university,
S.T.Hindu College,
Abishekapatti, Tirunelveli – 627012.

Abstract *– The security and privacy issues regarding outsourcing data have risen significantly as cloud computing has grown in demand. Consequently, since data management has been delegated to an untrusted cloud server in the data outsourcing phase, data access control has been identified as a major problem in cloud storage systems. To overcome this problem, in this paper, the access control of cloud storage using an Attribute-Based Access Control (ABAC) approach is utilized. First, the data must be stored in the cloud and security must be strong for the user to access the data. This model takes into consideration some of the attributes of the cloud data stored in the authentication process that the database uses to maintain data around the recorded collections with the user's saved keys. The clusters have registry message permission codes, usernames, and group names, each with its own set of benefits. In advance, the data should be encrypted and transferred to the service provider as it establishes that the data is still secure. But in some cases, the supplier's security measures are disrupting. This result analysis the various parameters such as encryption time, decryption time, key generation time, and also time consumption. In cloud storage, the access control may verify the various existing method such as Ciphertext Policy Attribute-Based Encryption (CP-ABE) and Nth Truncated Ring Units (NTRU). The encryption time is 15% decreased by NTRU and 31% reduced by CP-ABE. The decryption time of the proposed method is 7.64% and 14% reduced by the existing method.*

Keywords*: Cloud computing, data access control, Nth Truncated Ring Units, Ciphertext Policy Attribute-Based Encryption, database authentication.*

1. INTRODUCTION

The term "cloud computing" represents the supply of computational services on demand, primarily the collection of information and processing capacity [1]. This concept is commonly used to identify data centers that are accessible to multiple users on the Internet, without the user actively managing them [2]. Data is being transferred by an increasing number of businesses and individuals, personal data, and vast archive systems to cloud-based storage services because they provide a range of attractive services, such as limitless space, straightforward costs, and longstanding services. [3]. Consumers can also access applications and services without location limitations. However, according to several recent reports, 88% of cloud users are disturbed by the confidentiality of their information, and protection is frequently cited as the primary reason for using cloud-based storage solutions [4]. Cloud computing is a knowledge that allows Cloud storage service providers (CSP) to offer applications, calculate, and collection of information to customers located all over the world [5].

It has lately piqued the attention of both IT firms and academic organizations. In cloud computing, there are three major service delivery models: (PaaS) Platform as a Service, (IaaS) Infrastructure as a Service, and (SaaS) Software as a Service [6]. Private cloud, public cloud, community cloud, and hybrid cloud are the four types of cloud. Agility, flexibility, scalability, pay-per-use, and resiliency are only a few of the benefits of cloud computing. Scaling, low information technology costs, reliability, market stability, and almost unlimited efficiency are the advantages of cloud computing [7]. It has two

major data protection and access control issues, with data security weakening when reviewing its web-based services [8]. An access management model is a method for a user to gain access to data stored on cloud servers [9]. With the exponential development of big data technology and cloud computing, a growing number of enterprises and organizations have opted to automate their information to the server [10]. The majority of cloud data, like confidential medical history and business internal data, are extremely vulnerable [11].

The information would be maintained on the public cloud throughout the context of ciphertext in particular to provide data confidentiality and user privacy [12]. The encryption technique can be thought of as a protection assurance for gaining data access control. However, controlling access to encoded information is a significant problem [13]. Through the increasing adoption of cloud computing, increasingly consumers are opting to offload both the high responsibility for data processing and the complexity of computing to the public cloud [14]. About the benefits of cloud storage, secure information access management maintains among the most challenging obstacles, since the private cloud is not completely accepted via the data owner, and data collected in the cloud may contain sensitive data [15]. As a consequence, since distributing information to the server, the data owner should encrypt the message to preserve the safety of the customer and maintain secure communications. Here, Attribute-based access control is utilized to access the data in the cloud storage [16]. The remaining part of the paper contains section 2 explains the related work in various techniques and problems, and section 3 provides the proposed methodology and the step-by-step procedure of ABAC. Section 4 explains the result part and section 5 contains the conclusion parts.

2. LITERATURE REVIEW

There has been a lot of research on the different access controls in cloud storage. This section includes a discussion of the relevant work on access control.

A well-organized EACAS (attribute-based access control with an authorized search scheme) has been established by Jialu Hao *et al(2019)* [17] for the cloud storage access control. In the intended strategy, EACAS enables data users to customize search strategy with a focus on their data access and accumulate the respective trapdoor by using a private key conferred by the cloud provider to extract their valuable research by incorporating the key delegation methodology into AKP-ABE. But the limitation includes further modulation of the proposed methodology with supple exchanges of information with confined retaining of data in the cloud.

In 2019, Wang, S., et al, [18] analyzed a secure cloud storage framework. In this article, Ethereum blockchain architecture was used to construct a modern secure cloud storage framework including authentication, which was a mixture of Ethereum blockchain and CP-ABE. There was no trustworthy third party in the cloud computing system because it was decentralized. It has three features: it was built using Ethereum blockchain technologies, the storage operator can establish legitimate information usage times, and it can be preserved in the blockchain.

In 2018, Xu, Q., et al. defined that In a multi-authority cloud storage system, PMDAC-ABSC is a privacy-preserving shared data management mechanism based on Ciphertext-Policy ABSC that offers fine-grained control mechanisms and attributes privacy security at the same time [19]. The overhead decryption for users has been substantially reduced via outsourcing the unnecessary bilinear pairing to the cloud server without damaging the privacy of the attributes. The standard model is robust and can include anonymity, unforgeability, confidential authentication, and public verifiability. Their architecture would match protection goals towards practical computational efficiency, as demonstrated by the protection strategy asymptotic complexity comparison and execution outcomes.

In 2017, Liu, H., et. al,[20] implemented a logical secret sharing reward exchange mechanism, and a fair information access control system for data storage. The scheme produces a huge amount of fake keys. When a consumer deviates from the specified scheme during a share exchange, he or she must first send his or her shares. This discourages users from being narcissistic and encourages them to use the shared data as a community. According to mathematical research, the suggested scheme's Nash equilibrium is that both users still give their shares, enabling them to reconstruct the decoding key fairly. Furthermore, extensive research shows that the proposal will successfully control access control policies.

H-KCABE in data storage with fine-grained access control was developed by Sangeetha, M., et al [21]. In the HABE model, they propose an H-KCABE encryption algorithm with a few minor changes to improve performance through the re-encryption process. The HABE model helps the users to access information hierarchically through generating traffic, and the KCABE methodology improves efficiency by decreasing data transmission time in a fine-grained authentication method. They can easily improve efficiency by reducing time with the KCABE algorithm then the HABE model, which allows them to access information in a hierarchical manner without creating any traffic between users.

A new approach that resolves the essential encryption issue while also allowing for quick user voiding retraction has been employed by Zhihua, Liangao, and Dandan (2016) [22]. First, an access regulator is added to the current strategy, and so the attribute authority and authorization controllers create encryption data on a corporate level. Second, a version key that enables forward and reversible security is used to provide a convenient revocation process. The proposed method is simple and reliable in terms of user authorization and

revocation, according to the assessment. But lack of accuracy in terms of encryption of cloud storage

Saravanan, N., and Umamakeswari, D. A. [23] suggested a layered method to protecting client information that includes lattice-based encryption strategies It has been shown that by combining an access management architecture with a double authentication strategy, cloud data can be better protected. Users will be able to store their vast quantities of personal data in the cloud without fear of security threats thanks to this strong protection technique. The RSA and AES algorithms prevent the operator from guessing the key and encrypted text. Intruders' intelligence was almost irrelevant in terms of the hybrid paradigm. Bell and LaPadula (BLP) and lattice versions add user-level authentication as well.

In 2020 Challagidad, P. S., & Birje, M. N. [24] proposed an effective multi-authority intrusion detection system that enables efficient, fine-grained user authentication utilizing an attribute-based encryption scheme. For information storage anonymity, multi-authority access management, and fine-grained accessibility to encrypted information, the scheme uses HAS algorithm and a single RHA. The (RHA) Role Hierarchy Algorithm separates cloud users into groups depending on their assigned attributes. The (HAS) Hierarchy Access Structure assists in determining the authorization process for fine-grained and multi-authority cloud resource access management. In comparison to current works, analysis findings indicate that the RHA, HAS, was successful. Because more information is deposited on the cloud computing server, the scheme's advantages are growing increasingly obvious.

In cloud services, revocable server identity-based encryption for secure shard data was developed by Vurukonda, N., et al [25]. This paper explains revocable storage Identity-Based Encryption, a device that manages authenticated text back-and-forth authentication through disabled user revocation and software maintenance authentication functionality. Furthermore, the revocable storage IBE was compared to previous IBE approaches, demonstrating the reliability and sufficiency of the enables.

In 2019 Prabhu kavin, B., & Ganapathy, S. [26] proposed the latest data management method built on the Chinese Remainder Theorem (CRT) for safely processing user data in a cloud database. In addition, CRT was used to build a new community key management scheme for accessing encrypted data from the cloud database. In CRT-based secure processing systems, two encryption techniques were introduced using new methods for first and second authentication, as well as the formula for data storage authentication. In comparison, during the group key generation process formula for obtaining authenticated cloud data from a database server on a cloud server was introduced. By evaluating the experimental effects, the safeguards models' performance level has been assessed. Finally, the data protection model is superior to other current models.

3. PROPOSED METHOD

This section describes the access control in cloud storage using attribute-based access control. First, the data must be stored in the cloud and security must be strong for the user to access the data. This paradigm takes into account some of the features of cloud data seen in the database's security process for storing information about registered groups and the user's stored keys. Clusters, the registry's message identification code, and usernames and party names, each with their package of benefits. Initially, information must be secured and delivered to the service provider; this indicates that the data is protected if the supplier's security procedures are disrupted in some cases. The overall diagram of the design is given below.

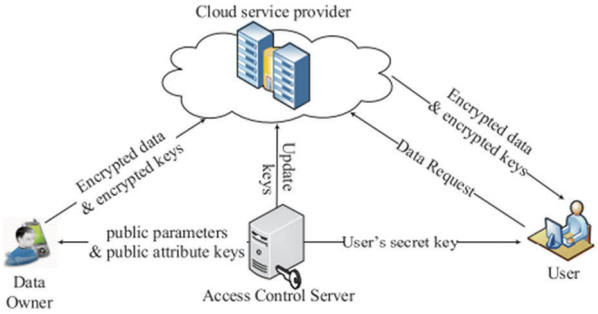

Fig. 1. Overall diagram of the proposed method

Data owner (DO): After transferring information to the server, DO must be authenticated with data based on its features, which expands user access to information based on their computer locations or passwords rather than data characteristics. DO has total trust in our method and is in charge of main development.

Data user (DU): The user is allowed to decode ciphertexts whose characteristics comply with DU's information system. It can also set a tighter search policy than his entry policy, and only his hidden key can be used to create the trapdoor. DU uses the cloud server's trapdoor to request the relevant data to extract the ciphertext that matches the search strategy. It is untrustworthy, and they can band together to procure data information outside their access rights. They're still curious about the data's attribute detail.

Cloud server (CS): CS is believed to have a lot of storage and processing power and is still available to help. The CS contains two parts: the (CSS) cloud storage server and the (DSS) delegated search server, with CSS supporting, DO in storing their information and DSS conducting data searches on behalf of DU and returning the related data to Data user. Cloud server is semi-honest, which ensures it would diligently comply with DO and DU's demands, but it is interested in data details, such as data content and attribute privacy. The Additional Private Key DSS is used to ensure that those without a private key are unable to guess the attribute values in the dropout by guessing offline.

A following objectives should be contacted when managing access to cloud storage.

Fine-grained access control: An information stored in the CSS is authenticated using its attributes, which can be decrypted via Data user if the ciphertext attributes obey the access policy. The access control should be built into the decoding mechanism rather than being handled by CS. Consequently, any threshold gate with an articulate information system should be enabled to ensure fine-grained network access.

Flexible and authorized search: DU must be allowed to obtain the information ciphertext whose attributes fulfill the selection policy using DSS. DU, on the other hand, can only scan the information inside the limits of his security authorization which ensures it must be allowed to provide a trapdoor with an exploration strategy that is more stringent than his information system. At the same time, the selection strategy must be expressed in a way that allows for an agile search.

Attribute privacy preservation: In ciphertext and trapdoor, the default attribute name is visible, but the associated component attributes should be concealed to secure sensitive information and privacy protection. Attribute values found in the ciphertext cannot be deduced by an attacker. Furthermore, any attackers who do not have the DSS private key are not exposed to attribute values in the search policy by the trapdoor.

Practical implementation: For functional implementations, device processes can be performed with lower computing and processing expenses.

3.1 OVERVIEW OF ABAC

ABAC is used to describe descriptive security policies for the DU and to explicitly encrypt attribute values in ciphertext which allows for better and more privately controlled access to outsourced data. The secret attribute knowledge, on the other hand, makes data search a difficult issue. ABAC's key delegates adopt a strategy that allows the DU to identify a more stringent search policy than the access system and use encryption data to create the next dropout to solve the issue. Figure 2 represents the overview of the ABAC method.

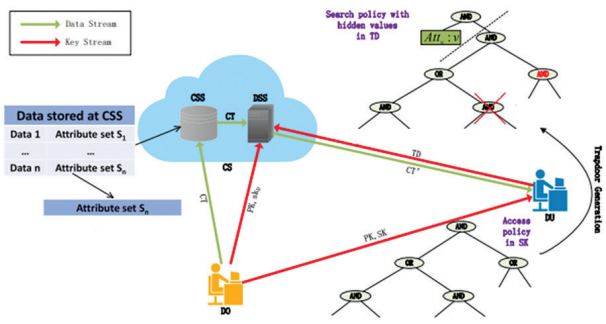

Fig. 2. Comprehensive operating procedures of ABAC

To protect the attribute information, the attribute values in the trapdoor are also concealed. A synthetic attribute on both the ciphertext then the trapdoor prevents DSS from accessing the data content. In detail, the ciphertext is made up of two sections: (1) The actual attribute set is used to encrypt the initial data; (2) a meaningless data "1" encoded with the synthetic attribute inserted through the original attribute set. The virtual characteristic is connected to the search tree root node with an AND gate while in the trapdoor, allowing it a prerequisite for successful matching. DSS will decrypt the ciphertext which descriptor set supports the search policy by deciding whether the trivial data "1" can be obtained by checking the ciphertext, but it cannot decrypt the ciphertext of the original data that is encrypted without the virtual attribute. As a consequence, information security will achieve fine-grained access control with an agreed-upon search on information outsourced to the cloud while maintaining data integrity and attribute privacy.

3.2 Step by step procedure of ABAC2

ABAC consists of six phases: data encryption, system setup, data decryption key generation, trapdoor generation, and data search.

3.2.1 System Setup

To produce *PK* (Public key) and *MSK (master secret key)*, *DO* choose a security limitation ξ and call the Setup (ξ) algorithm. The Setup algorithm is similar to ABE, with the exception that the public key includes a virtual attribute V_a containing the value *v* which is the value of real attributes, and additio nal public and private key pair (pk_D, sk_D) for DSS is created as $pk_D = g^y$, and $sk_D = \gamma$, where γ is a random value in Z^*_p. The system's public key is then made available as,

$$PK = \langle g, u, h, w, e(g,g)^{\alpha}, g_1, g_2, g_3, g_4, [V_a : v], pk_D{}^1 \rangle \quad (1)$$

DO maintains the machine master secret key as $MKS = \langle \alpha, \tau_1, \tau_2, \tau_3, \tau_4 \rangle$. DO also passes the private key $sk_D = \gamma$ to the DSS.

3.2.2 Key generation

The *DSS* public key pk_D is used in the machine public key *PK* for convenience. *DO* creates an access policy *AP* for *DU* based on his position and distributes the hidden key $SK = \langle AP, \{D_x, D_{x,0}, D_{x,1}, D_{x,2}, D_{x,3}, D_{x,4}\}_{n_x \in atts(\tau)} \rangle$ created by the *KeyGen(PK, MSK, AP)* algorithm to *DU* when he enters the framework. The *ABE* and *KeyGen* algorithms are the same to generate keys for public and private keys. *DU* will decode the ciphertext whose attribute collection satisfies *AP* using *KeyGen* and the secret key *SK*

Data Encryption

DO creates a characteristic set *S* based on the information specifications before transferring the data *M* to *CS*, and then uses the *Encrypt(PK, M, S).* process to generate the ciphertext *CT*. *DO* computes $E=M.e(g,g)^{as}$, $E=g^s$,

$E' = g^{s'}$, two random values s and s'. Then take specific datatypes $s(x,1)$, $s(x,2)$, and $s_{x,1}, s_{x,2}, z_x$ from z^*_p for each component in S are chosen and computed.

$$E_{x,0} = w^{-s}(u^{sx}h)^{zx}, E'_{x,0} = w^{-s'}(u^{sz}h)^{zx} \quad (2)$$

$$E_{x,1} = g_1{}^{zx-8x,1}, E_{x,2} = g_2{}^{sx,1}, \quad (3)$$

$$E_{x,3} = g_3{}^{zx-8x,2}, E_{x,4} = g_4{}^{s_{x,2}} \quad (4)$$

DO selects a random value $r_v \in Z^*_p$ for the virtual attribute v_a and quantifies $E_{v,0} = w^{s'}(u^v h)rv$, $E_{v,1} = g^{rv}$. Lastly, the ciphertext that will be uploaded to the cloud is developed as follows:

$$CT = \langle N_S, \tilde{E}, \widetilde{E'}, E, E', E_{v,0}, E_{v,1}, \{E_{x,0}, E'_{x,1}, E_{x,1}, E_{x,2}, E_{x,3}, E_{x,4}\}_{x \in S} \rangle \quad (5)$$

3.2.3 Trapdoor generation

DU establishes a *SP* development scheme based on user access policy, in which search policy *(SP)* will have the same expression style as access policy *(AP)*, the search tree architecture in search policy is extremely strict than the request tree architecture in *AP*, and the meaning of the element related to the attribute name cannot be altered. Then, using his hidden key *SK* identified through the data access access policy, *DU* uses the *TrapGen(PK, SK, SP)* algorithm to produce the trapdoor TD identified with the search policy search policy. The *TrapGen* algorithm uses the key delegation method, in which a sequence of simple operations is carried out to transform the hidden key *SK* for the efficient key access policy to the trapdoor for the search policy search policy. The *TrapGen* algorithm includes the corresponding three steps in particular. The first step is to manipulate the current gates to convert the actual private key to a different encryption key, the second option is to prevent DSS from decoding the information to the encrypted message via adding an AND gate to the root node, then the final step is to protect the related data in the trapdoor against disconnected manipulation attacks by attackers who do not have access to the DSS private key sk_D.

3.2.3 Data decryption

DU uses the *Descrypt (CT', SK)* algorithm to retrieve the received information since obtaining the encrypted message from DSS. With increasing attribute name $n_x \in \widetilde{\mathcal{N}_l}$ in the Decrypt algorithm computes,

$$P_x = e(E, D_x)e(E_{x,0}, D_{x,0}).(Q_x)^{1/\delta_y} = e(g,g)^{px(0)s} \quad (6)$$

In the *ABE.Decrypt* algorithm, the term $(g,g)^{\alpha s}$ can be improved and M can be determined concluded $E/_{e(g,g)^{\alpha s}}$ The encrypting data the product supplier is assigned to assures that the data is still secure in case the supplier's security measures are violated. After that, the data can be accessed by the user using an encrypted key.

4. RESULT AND DISCUSSION

In this section, the access control in cloud storage is analyzed using ABAC. The proposed methodology is applied in the JAVA programming language with JDK 1.7.0. This proposed concept is mainly used in the health care system. The experimental used datasets are collected from different sources.

4.1 COMPARATIVE ANALYSIS

A proposed method is analyzed via several methods like key generation, encryption time, time consumption and decryption time. A current method is investigated against the existing methods are NTRU and CP-ABE. The proposed method comparative analysis against the existing technique is given below,

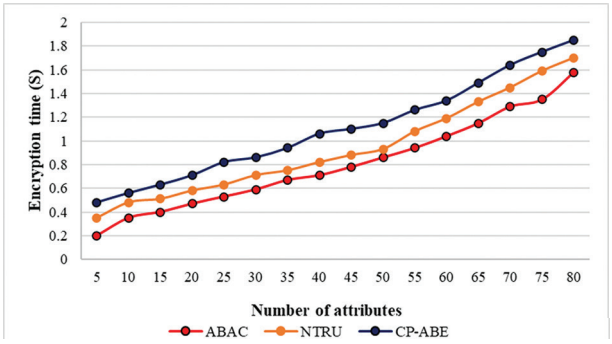

Fig. 3. Comparative analysis of encryption time

In Figure 3, represents the comparative analysis of ABAC, NTRU, and CP-ABE with encryption time and several attributes. The encryption time is the amount of time it takes for an encryption algorithm to generate a ciphertext from plaintext and it is used to measure the performance of the encryption scheme. Each attribute in the encryption process (ABAC, NTRU, and CP-ABE) can begin at the same attribute value, but they can vary by changing the values of the time representation. The encryption time may increase which indicates the speed of the encryption process. The encryption time of the proposed method is 15% decreased by the existing method NTRU and 31% of the encryption time is reduced by the existing method CP-ABE. The graphical representation of the key generation is given below,

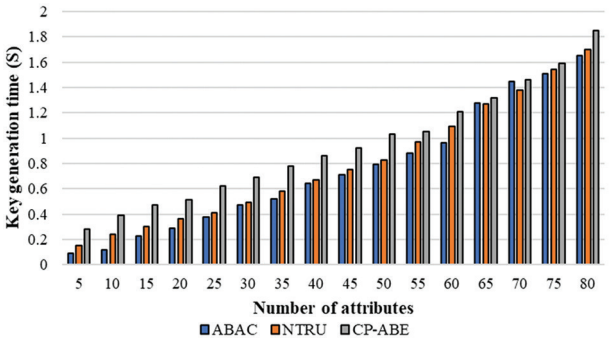

Fig. 4. Comparative analysis for Key Generation Time

In Figure 4, represents the comparative analysis of ABAC, NTRU, and CP-ABE with key generation time and a number of the attribute. The key generation time can also increase the variance of attributes, the starting stage of attribute values may same but the variations of keys may differ. The ABAC is lower compared to other graphical representations, NTRU may be slightly higher compared to ABAC and CP-ABE are higher values in key generation performance. The proposed method key generation is 6.16% reduced by the existing method NTRU and 22% decreased by CP-ABE. The decryption time graph is given below,

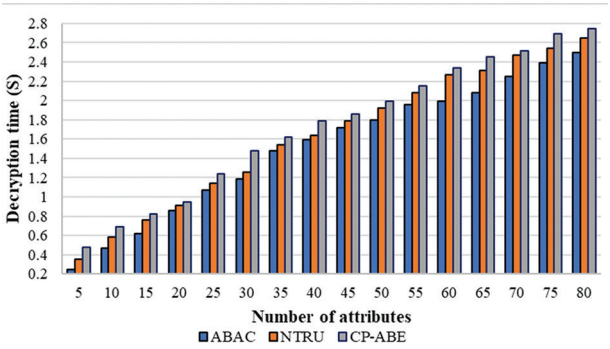

Fig. 5. Comparative Analysis for decryption time

In figure 5, represents the comparative analysis of ABAC, NTRU, and CP-ABE with decryption time and number of the attribute. In a decryption time process, the ABAC attributes may be very less compared to other attribute representations and the other attribute value may increase step by step. The decryption time of the proposed method is 7.64% and 14% reduced by the existing method. The graphical representation of time consumption is given below,

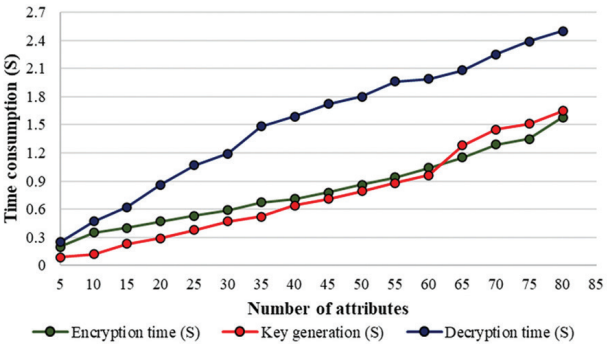

Fig. 6. Comparative analysis for time consumption with encryption time, key generation, decryption time

In figure 6, represents the comparison analysis of time consumption with encryption time, key generation, and decryption time. The encryption time may increase slightly throughout the key generation, the key generation may also increase but it can decrease towards the encryption process. The encryption time and key generation are mixed while increases neither decrease. The decryption time can increase highly by comparing the other two comparisons.

5. CONCLUSION

In this section, we have introduced the access control in cloud storage data using ABAC. First, the data can be stored in the cloud and security must be strong for the user to access the data. This model considers some of the characteristics of the cloud data contained in the authentication mechanism that the database uses to retain data around groups that have been registered, as well as the user's saved keys. User names and party names, as well as groups and the database message encryption method all, have unique benefits. Encrypting the data before sending it to the network operator means that, it remains encrypted despite the supplier's protection protocols being breached. The suggested method's experiment results are assessed utilizing a variety of metrics, including encryption time, decryption time, key generation time, and time usage. The encryption time of the proposed method is 15% decreased by the existing method NTRU and 31% of the encryption time is reduced by the existing method CP-ABE. The decryption time of the proposed method is 7.64% and 14% reduced by the existing method. The key generation of the proposed method is 6.16% reduced by the existing method NTRU and 22% decreased by CP-ABE. By comparing the time consumption, the key generation time is reduced.

6. REFERENCES:

[1] J. Li, W. Yao, Y. Zhang, H. Qian, J. Han, "Flexible and Fine-Grained Attribute-Based Data Storage in Cloud Computing", IEEE Transactions on Services Computing, Vol. 10, No. 5, 2017, pp. 785–796.

[2] J. Shi, J. Lai, Y. Li, R. H. Deng, J. Weng, "Authorized Keyword Search on Encrypted Data", Proceedings of the European Symposium on Research in Computer Security, Wroclaw, Poland, 7-11 September 2014, pp. 419–435.

[3] P. Jiang, Y. Mu, F. Guo, Q. Wen, "Public Key Encryption with Authorized Keyword Search", Proceedings of the Australasian Conference on Information Security and Privacy, 2016, pp. 170–186.

[4] H. Cui, Z. Wan, R. H. Deng, G. Wang, Y. Li, "Efficient and Expressive Keyword Search Over Encrypted Data in The Cloud", IEEE Transactions on Dependable and Secure Computing, Vol. 15, No. 3, 2016, pp. 409–422.

[5] H. Cheng, C. Rong, K. Hwang, W. Wang, Y. Li, "Secure Big Data Storage and Sharing Scheme For Cloud Tenants", China Communications, Vol. 12, No. 6, 2015, pp. 106–115.

[6] J. Baek, Q. H. Vu, J. K. Liu, X. Huang, Y. Xiang, "A Secure Cloud Computing-Based Framework for Big

Data Information Management of Smart Grid", IEEE Transactions on Cloud Computing, Vol. 3, No. 2, 2015, pp.233–244.

[7] G. Zhuo, Q. Jia, L. Guo, M. Li, P. Li, "Privacy-Preserving Verifiable Set Operation in Big Data for Cloud-Assisted Mobile Crowdsourcing", IEEE Internet of Things Journal, Vol. 4, No. 2, 2016, pp. 572–582

[8] L. Guo, Y. Fang, M. Li, P. Li, "Verifiable Privacy-Preserving Monitoring for Cloud-Assisted M-health Systems", Proceedings of the IEEE Conference on Computer Communications, 26 April - 1 May 2015, pp. 1026–1034.

[9] L.-Y. Yeh, P.-Y. Chiang, Y.-L. Tsai, J.-L. Huang, "Cloud-Based Fine Grained Health Information Access Control Framework for Lightweight IoT Devices with Dynamic Auditing and Attribute Revocation", IEEE Transactions on Cloud Computing, Vol. 6, No. 2, 2018, pp. 532–544.

[10] Z. Yan, X. Li, M. Wang, A. V. Vasilakos, "Flexible Data Access Control Based On Trust And Reputation In Cloud Computing", IEEE Transactions On Cloud Computing, Vol. 5, No. 3, 2017, pp. 485–498.

[11] K. Yang, K. Zhang, X. Jia, M. A. Hasan, X. Shen, "Privacy-Preserving Attribute-Keyword Based Data Publish-Subscribe Service on Cloud Platforms", Information Sciences, Vol. 387, 2017, pp. 116–131.

[12] J. Hao, C. Huang, J. Ni, H. Rong, M. Xian, X. Shen, "Fine-Grained Data Access Control with Attribute-Hiding Policy for Cloud-Based IoT", Computer Networks, Vol. 153, 2019, pp. 1-10.

[13] H. Cui, R. H. Deng, G. Wu, J. Lai, "An Efficient and Expressive Ciphertext-Policy Attribute-Based Encryption Scheme with Partially Hidden Access Structures", Proceedings of the International Conference on Provable Security, 2016, pp. 19–38.

[14] R. Fernando, R. Ranchal, B. An, L. Othmane, B. Bhargava, "Consumer Oriented Privacy Preserving Access Control of Electronic Health Records in The Cloud", Proceedings of the IEEE 9th International Conference on Cloud Computing, San Francisco, CA, USA, 27 June- 2 July 2016, pp. 608–615.

[15] R. Ranchal, B. Bhargava, R. Fernando, H. Lei, Z. Jin, "Privacy Preserving Access Control in Service-Oriented Architecture," Proceedings of the IEEE International Conference on Web Services, San Francisco, CA, USA, 27 June - 2 July 2016, pp. 412–419.

[16] Z. Wang, D. Huang, Y. Zhu, B. Li, C.-J. Chung, "Efficient Attribute-Based Comparable Data Access Control", IEEE Transactions on Computers, Vol. 64, No. 12, 2015, pp. 3430–3443.

[17] J. Hao, J. Liu, H. Wang, L. Liu, M. Xian, X. Shen, "Efficient Attribute-Based Access Control with Authorized Search in Cloud Storage", IEEE Access, Vol. 7, 2019, pp.182772–182783.

[18] S. Wang, X. Wang, Y. Zhang, "A Secure Cloud Storage Framework with Access Control based on Blockchain", IEEE Access, Vol. 7, 2019, pp. 112713–112725.

[19] Q. Xu, C. Tan, Z. Fan, W. Zhu, Y. Xiao, F. Cheng, "Secure Multi-Authority Data Access Control Scheme in Cloud Storage System Based on Attribute-Based Signcryption", IEEE Access, Vol. 6, 2018, pp. 34051–34074.

[20] H. Liu, X. Li, M. Xu, R. Mo, J. Ma, "A Fair Data Access Control Towards Rational Users In Cloud Storage", Information Sciences, Vol. 418, 2017, pp. 258–271.

[21] M. Sangeetha, P. Vijayakarthik, S. Dhanasekaran, B. S. Murugan, "Fine Grained Access Control Using H-KCABE in Cloud Storage", Materials Today: Proceedings, Vol. 37, 2021, pp. 2735–2737

[22] Z. Xia, L. Zhang, D. Liu, "Attribute-based Access Control Scheme With Efficient Revocation In Cloud Computing", China Communications, Vol. 13, No. 7, 2016, pp.92–99.

[23] N. Saravanan, D. A. Umamakeswari, "Lattice Based Access Control for Protecting User Data in Cloud Environments with Hybrid Security", Computers & Security, Vol. 100, 2021, p.102074.

[24] P. S. Challagidad, M. N. Birje, "Efficient Multi-authority Access Control using Attribute-based Encryption in Cloud Storage", Procedia Computer Science, Vol. 167, 2020, pp. 840–849.

[25] N. Vurukonda, M. T. Basu, V. Velde, K. Enumula, "Revocable Storage Identity-Based Encryption For Protected Shared Data In Cloud Computing", Material Today: Proceedings, 2020

[26] B. P. Kavin, S. Ganapathy, "A Secured Storage and Privacy-Preserving Model Using CRT for Providing Security on Cloud and IoT-Based Applications," Computer Networks, Vol. 151, 2019, pp.181–190.