**[1]Sachin Popat Patil**

**[2]Mustafa Basthikodi**

**[3]Kumaraswamy S**

**[4]Ananth Prabhu Gurpur**

**[5]Akashraj Raga**

# Enhancing Data Privacy Protection in Cloud Computing Through Ciphertext-Policy Attribute-Based Encryption

Journal of Electrical Systems

***Abstract: -*** The paper highlights the significance of Ciphertext-Policy ABE in enabling broadcast, secure transmission, and effective control over data access within cloud computing systems. However, the existing implementation of Ciphertext-Policy ABE presents a vulnerability, as plaintext access policies transmitted alongside ciphertexts may unintentionally expose sensitive user and data privacy information. To address this issue, a novel technique is proposed, which leverages hashing techniques to conceal access policies and strengthens protection against potential attacks through signature validation methods. Comparative analysis with existing techniques emphasizes the effectiveness of the proposed scheme in enhancing privacy and security within cloud computing environments. By addressing vulnerabilities in Ciphertext-Policy ABE and introducing innovative methods for access policy concealment and security enhancement, this study contributes to advancing secure data management practices and promoting greater confidence in cloud computing systems.

***Keywords:*** cloud security, Ciphertext-Policy ABE, Key-Policy ABE, Access control, Privacy

## I. INTRODUCTION

The accessibility and cost-effectiveness of distributed storage services drive organizations and individuals to adopt them, fueled by the abundance of data from sources like cloud-enabled applications and the Internet of Things (IoT). However, entrusting data to untrusted cloud servers raises security concerns. Cloud service providers (CSPs) may genuinely store data but seek to extract as much information as possible from interactions with clients, posing risks to data privacy. Encrypting data before transferring it to the cloud can mitigate these risks, but a clear encryption strategy is needed to prevent information disclosure and maintain control over data. Security issues, particularly in IoT-based applications, prompt researchers to develop modern security mechanisms, focusing on user access controls and maintaining data protection in the cloud.[1]

Cloud computing is a prominent IT advancement attracting attention from industry and government. Despite the benefits of cloud services, ensuring robust security and storage systems remains challenging due to the web-based nature of storage and data association. The vast amount of confidential and sensitive information stored in the cloud necessitates higher security mechanisms, including robust authentication techniques.

Before migrating to the cloud, client information must be encrypted to enhance information security. Various encryption techniques, such as symmetric and asymmetric encryption, are proposed, but key management remains a challenge. Generating a private key for each client in the information sharing system helps overcome this challenge, preventing data leakage and ensuring enhanced security.

Attribute-based encryption (ABE)[2] addresses access control challenges. Key-Policy ABE (KP-ABE) [3] and Ciphertext-Policy ABE (CP-ABE) are the primary methods. CP-ABE [4], associating attributes with client keys and

[1] Research Scholar, Department of Computer Science & Engineering, Sahyadri College of Engineering & Management, Mangaluru, India, Email: sachinpatil.it@gmail.com

[2*] Department of Computer Science & Engineering, Sahyadri College of Engineering & Management, Mangaluru, India, Email: mbasthik@gmail.com

[3] Department of Computer Science & Engineering, UVCE, Bengaluru, India, Email: kumar.aruna@gmail.com

[4] Department of Computer Science Engineering, Sahyadri College of Engineering & Management, Mangaluru, India, Email: educatorananth@gmail.com

[5] Associate Research Analyst, Tor Secure, Mangaluru, India, Email: akash.raga@gmail.com

access policies with ciphertext, offers more effective access control. However, embedding access policies within ciphertexts increases the risk of insider attacks. Verifying information owner's authentication is crucial. [5] Data protection and security in cloud computing remain vital academic topics, with studies identifying key security attributes and exploring various secure communication techniques. Figure 1 illustrates the risks and associated technologies in cloud computing. [6][7]

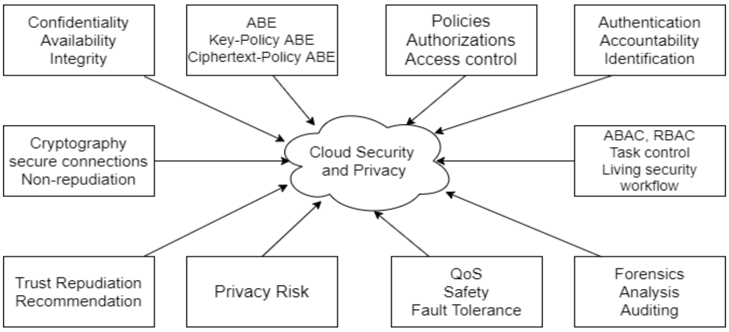

**Figure 1 Cloud Computing Risks and Associated Technologies**

Securing effective access control of information and resources is paramount in cloud computing. [8] Once the true identity of the user is established, the access control system limits the capabilities and permissions of the requester to access the data. This measure is typically employed to safeguard critical data assets and deter unauthorized activity by legitimate users. [9]

**Table 1 Comparisons of general Models of access control [10]**

| | ABAC | TBAC | RBAC | MAC | DAC | UCON |
|---|---|---|---|---|---|---|
| Confidentiality | NO | NO | NO | YES | YES | NO |
| Safety | NO | NO | NO | YES | NO | YES |
| Flexibility of Authority | YES | YES | YES | NO | YES | YES |
| Minimum Privilege | YES | YES | YES | YES | NO | YES |
| Duty separation | YES | YES | YES | YES | NO | NO |
| Fine grain | YES | YES | NO | YES | YES | YES |
| Description of Constraint | YES | NO | YES | YES | NO | YES |
| Dynamic | YES | YES | NO | NO | YES | YES |
| Compatibility | YES | NO | YES | NO | YES | YES |
| Ease of Management | NO | NO | YES | YES | NO | NO |
| Ease of Modeling | YES | NO | YES | YES | YES | NO |
| Expansibility | YES | YES | NO | NO | YES | YES |
| Descriptive ability | YES | YES | YES | YES | YES | NO |

Various access control mechanisms play significant roles in ensuring data security, including Discretionary Access Controls (DAC), Usage Controls (UCON), Mandatory Access Control (MAC), Attribute-Based Access Controls (ABAC), Task-Based Access Controls (TBAC), and Role-Based Access Controls (RBAC). Table 1 presents a performance comparison of these technologies, featuring entries denoted by 'YES' and 'NO'. With the evolution of access control strategies, many models inherently incorporate relationship logic, as depicted in Figure 2. By implementing a set of rules and methodologies, access control enables genuine users to access different data securely, safeguarding data privacy, confidentiality, and integrity within network security. [11]

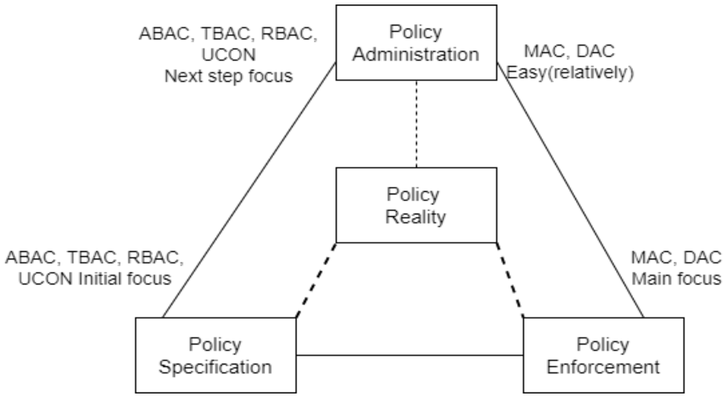
ABAC, TBAC, RBAC, UCON Next step focus
Policy Administration
MAC, DAC Easy(relatively)
Policy Reality
ABAC, TBAC, RBAC, UCON Initial focus
MAC, DAC Main focus
Policy Specification
Policy Enforcement

**Figure 2 Interconnections Among Access Control Models**

When accessing cloud computing services for computation and storage, clients must undergo validation by the Cloud Service Provider (CSP) and employ appropriate strategies to access data and services securely. Ensuring security in cloud computing necessitates mutual authentication and access control between CSPs, while cloud users must also guard against side-channel attacks and implement relevant measures to secure data. [12]

## II. ADVANCING DATA SECURITY WITH ATTRIBUTE-BASED ENCRYPTION

In the ABE framework, participating entities include users and authority agencies. Authorized agencies manage attributes and issue attribute keys to users. Users are categorized as message senders and recipients. The basic ABE model proposed by authors in [13] involves mapping each attribute in the system to $W_z$ (set {z =0..z-1} under modulo of multiplication of prime number) using a hash function. Both ciphertext and client keys are associated with attributes. This approach supports attribute-based threshold policies. For example, if a document in a library has attribute sets {technology, protection, kannada, engineer}, and the attribute encryption threshold is set to 2, users with attribute sets {technology, Kannada, engineer} can access the document, while those with attribute sets {Kannada, engineer} cannot.

The fundamental ABE technique comprises four phases: SettingUp, Key Generation, Encryption, and Decryption. While initializing, the framework runs as per the parameters of security and produces two groups $G_1$, $G_2$ with primary value $q$ and bilinear pairs $b$: $G_1 \text{ X } G_1 \rightarrow G_2$, $t$ is threshold value. The steps involved in the ABE technique are outlined below.

***Setting Up***: Make authorized agencies to randomly choose *x, p1, p2,..., pn* belongs to $W_z$, public key of system *PKey* is ($P_1 = g^{p1}$ ,…, $P_n = g^{pn}$ , $X = e(g,g)^x$), $(x, p_1, \ldots, p_n)$ is master key *MKey*.

***Key Generation***: Production of private key of customer c by authorized agency, choose randomly a polynomial *poly* of ($n$ - 1) degree, assume $poly(0) = x$, private key of customer *SKey* is $\{D_j = g^{poly(j)/pj}\}$ for every *j belongs to cypher text Attribute $C_a$.*

***Encryption***: Sending customer encrypt message along with attribute $C_a$, choose randomly k belongs to $W_z$, the ciphertext is ($C_a$, $F = X^k H = e(g, g)^{xk}$ H, $\{F_i = g^{pjk}\}$ for every j *belong to* $C_a$)$M$ belongs to group $G2$, where F is edges in the access graph, k is interior nodes fulfilling an access structure.

***Decryption***: Suppose Modulus of ($C_u \cap C_a$) > n, receiving customer can select n attributes, suppose j belongs to $C_u \cap C_a$, and calculate $e(F_j, D_j) = e(g, g)^{poly(j)k}$, obtain $H = F/X^k$ while obtaining $X^k = e(g, g)^{poly(0)k} = e(g, g)^{xk}$ by making use of Lagrange interpolation approach.

In the technique mentioned above, Key Generation step utilizes sharing mechanism of threshold secret, that installs secret x into every part Dj of S Key to carry out threshold strategy. S Key is connected with irregular polynomial poly, that makes it unimaginable for various clients to do conspiracy attacks with private keys. Encryption technique utilizes bilinear blending to encode messages, and ciphertext parts Fj are connected with attribute, consequently indicate the essential attributes for decoding. Arbitrary numbers k can keep clients from decryption of resulting cipher text effectively. In above basic ABE techniques, P Key is straightly connected with the quantity of framework attributes, and the quantity of power tasks and bilinear logarithms is higher.

Basic ABE can address limit boundary activity of attributes, and the limit boundary is set by the authorities. Numerous pragmatic applications required to support 'boundary limit(threshold), 'OR', 'AND' and non-activity of attributes as per adaptable access control procedures, so the source can determine access control systems.

ABE is a sort of open key encryption where the secrets key of a client and ciphertext are subject to attributes. In such a framework, decoding of cipher text is conceivable, provided that the arrangement of key attributed of client matches cypher text's attributes. Two kinds of encryption methods dependent on attributes are: Key-Policy ABE, as illustrated in Figure 3 and Cipher text-Policy ABE, as illustrated in Figure 4.

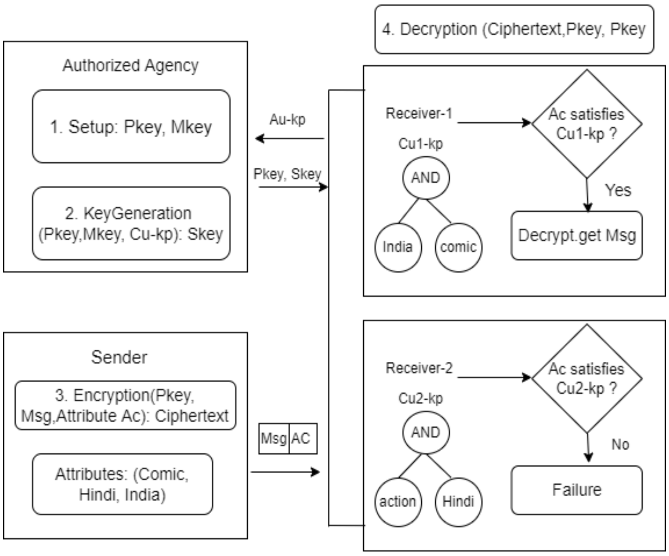

**Figure 3 Key-Policy ABE Technique [14]**

**Key-Policy ABE:** As the illustration of Key-Policy ABE in Figure 4, client's keys take on tree design to portray access strategy $C_u$-KP, the collection of leaf nodes of tree is $C_u$. Ciphertext is connected with set of attributes $C_a$, when $C_a$ fulfills $C_u$-KP, ciphertext will be decrypted by clients.

The distinction between Key-Policy ABE and basic ABE techniques are KeyGeneration and Decryption mechanisms. KeyGeneration technique can utilize secret sharing instrument and embrace hierarchical strategy to characterize an arbitrary polynomial polyn whose count of times is not exactly the threshold value of nodes for every node n in the tree.

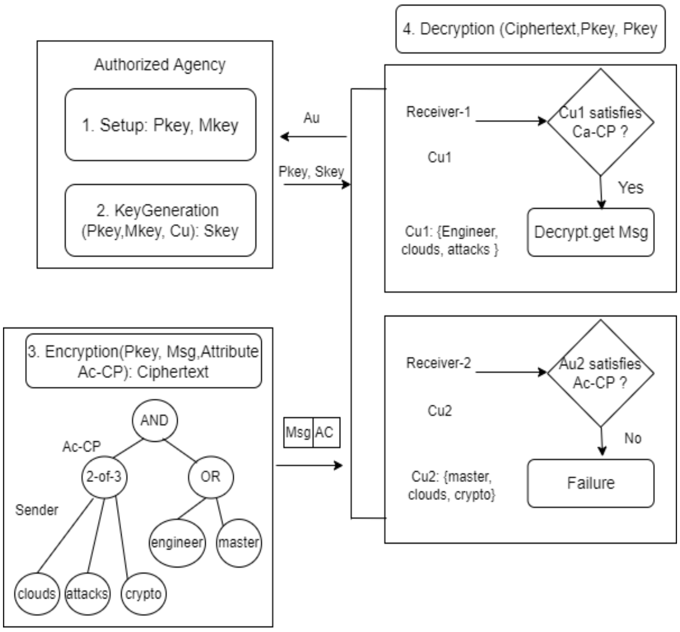

**Figure 4 Ciphertext-Policy ABE Technique [15]**

In the event that polyn(0) = $P_{parent(n)}$ (index(n)), and parent(n) addresses parent node of n, index(n) addresses the count of n indexes, while *rt* is root node, then, at that point, $p_{rt}(0) = x$, and expert key x is scattered into the private key part Di with respect to the leaf node.

The techniques used for decryptions decodes every node by recursive operation from base to top, and gets the secret values expected to recuperate plaintext. As demonstrated in Figure 4, $C_a$ fulfils strategy Cu1-KP, node collection V in decoding tree is [AND], cipher text embraces tree construction to depict the access strategy $C_a$-CP and accomplishes the entrance control strategy chose by the source. In Cypher text Policy-ABE, the client's key is connected with attribute set $C_u$, when $C_u$, fulfils $C_a$-CP, the client can decode the cipher text.

Unlike the fundamental ABE techniques, the length of P Key and M Key is autonomous of the quantity of framework attributes in the Cypher text Policy-ABE. Both Key Generation and Cypher text Policy-ABE utilize two-stage irregular mask to forestall collusion among clients. The client's private key is connected with the second-stage irregular number.

In technique of encryption, the execution of access tree is like Key Generation mechanism of Key-Policy ABE, thing that matters is $p_r(0) = x$, and the leaf node compares to the ciphertext part $F_j$. Decryption technique is same as Key-Policy ABE, yet the quantity of bilinear pairings is twice. In Figure 5, $C_{u1}$ fulfils $C_a$-CP, and the arrangement of interior nodes x in the tree is decoded.

Since the encryption rules are contained in the encryption mechanism, the expense has been incredibly decreased in network data transfer capacity and node handling overhead. In Key-Policy ABE, clients' secret keys are produced dependent on an access tree that characterizes the consent extent of significant clients, and information are encoded over a bunch of attributes.

Nonetheless, Ciphertext-Policy ABE utilizes access tree for encryption of information, and clients' secret keys are created over a collection of attributes. Access control strategy is related with cipher tree, the decoding keys limited by a bunch of attributes which can be depicted, and decoding keys can be gotten when the decrypting party own policies matched by attributes.

**Cypher text-Policy** ABE mechanism is very well known in cloud computing environment. To put it plainly, three ABE strategies are very unique, explicit details are displayed in Table-2, basic ABE can utilize boundary limit policy, the extent of utilization is generally restricted; both Key-Policy ABE and Ciphertext -Policy ABE have a lot more extensive applications, yet the weight of encrypting, decrypting and correspondence is extremely weighty.

Encrypting party has no compelling reason to realize who do decryption of the encrypted data, when the decrypting party can meet suitable strategy, which can perform decryption of the ciphertext. ABE mechanism depends on Bilinear blending hypothesis of elliptic curve, which is difficult to be interpreted. ABE is joined to an access structure in the security reproduction, which is hard to be implanted into a customary hard supposition in view of the complexity of access structure, hence it is extremely protected in principle and practice.

**Table 2 Key-Policy ABE and Ciphertext-Policy ABE comparisons**

| | | **Key-Policy ABE** | **Ciphertext-Policy ABE** |
|---|---|---|---|
| **SettingUp (λ, U)** | Input | Safety Parameters | Safety Parameters |
| | | Size of Attribute space | Size of Attribute space |
| | | Size of User space | Size of User space |
| | Results | Pkey | Pkey |
| **Encryption (Pkey, Msg, Attributes)** | Input | Mkey | Mkey |
| | | Pkey | Pkey |
| | | Message | Message |
| | | Set of Attributes | Structure of Access |
| | Results | Ciphertext | Ciphertext |
| **Key Generation (Mkey, S)** | Input | Mkey | Mkey |
| | | Structure of Access | Set of Attributes |
| | | Pkey | NA |
| | Results | Ciphertext | Secret Key |

| | | | |
|---|---|---|---|
| **Decryption (Pkey, Ciphertext, Skey)** | Input | Pkey | Pkey |
| | | Ciphertext | Ciphertext |
| | | Decryption key | Secret Key |
| | Results | Raw message | Raw message |

Researchers in [16] proposed a strategy dependent on Ciphertext-Policy ABE mechanism, which utilized intermediary encryption tool to help the repudiation of Key-Policy ABE, and the pertinent individual from master key was refreshed with the denial attribute, then, at that point, it produced another intermediary key to go along the attribute of the moment renouncement. Some authors also attempted to solve by merging the cloud security with parallel computing.[17][18]

The work at [19] proposed to add a period cut-off to the client's attribute, every attribute authority can powerfully erase any client from its space, and the individuals who are repudiated can't get to resulting re-evaluated information, the particular component is discussed in the next section. Likewise, in [20][21] proposed a attribute disavowal strategy dependent on Ciphertext-Policy ABE, CSP has a specific trust degree, the information owner would convey the CSP execution, however the access program structure tree just upheld the "AND", thusly, it can't give a fine grained and adaptable access control strategy.

## III. FRAMEWORK AND METHODOLOGY

The framework of proposed access control technique is outlined in Figure 5, which gives assurances to convey fine-grained access control alongside protection from insider's attacks. The framework comprises of four distinct elements.

*Information Owner* is responsible for encryption of all information utilizing access strategy prior to moving to the cloud. The *Cloud Server* stores the information owner's documents and permits customers with license to data access. *Customer*, key production system is responsible for generating a secret key for customers of the cloud. The genuine customer having secret keys fulfilling the policy of access is able for decryption of information. *Key Production System (KPS)* is responsible for production and distribution of secret keys to genuine customers of the cloud.

In the beginning, KPS produces a public key and a master key in initial stage. Then, KPS sends public key to information owner. The information owner performs encryption of information in stage-3.

Information owner moves information which are encrypted to cloud server in stage-4 with anonymously used policies for access. Then in stage-5, customer sends information request to cloud server. The stage-6 utilized for cloud server to send a ciphertext to customer of information. Then, in stage-7, information user places a request for secret keys for ciphertext received. In stage- 8, KPS gives response to information user with a secret key. At the end, in stage-9, information user performs decryption of ciphertexts and verifies for the authenticity of signature.

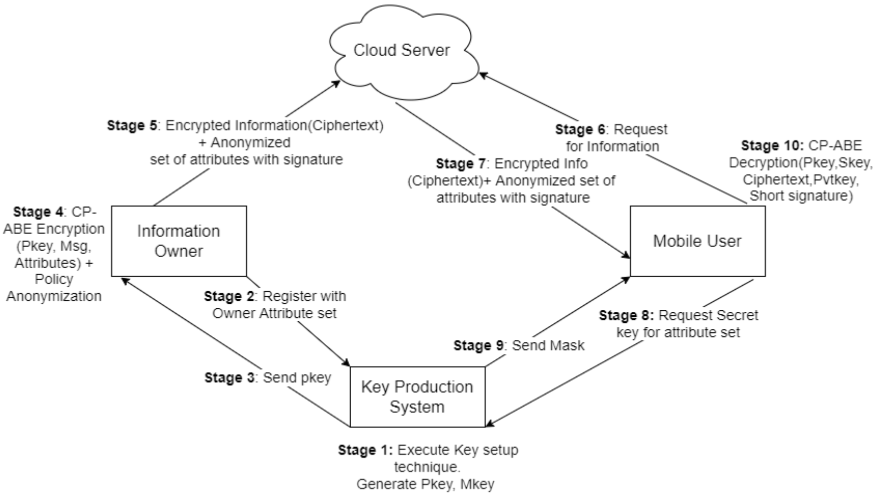
Cloud Server
Stage 5: Encrypted Information(Ciphertext) + Anonymized set of attributes with signature
Stage 6: Request for Information
Stage 7: Encrypted Info (Ciphertext)+ Anonymized set of attributes with signature
Stage 10: CP-ABE Decryption(Pkey,Skey, Ciphertext,Pvtkey, Short signature)
Stage 4: CP-ABE Encryption (Pkey, Msg, Attributes) + Policy Anonymization
Information Owner
Mobile User
Stage 2: Register with Owner Attribute set
Stage 8: Request Secret key for attribute set
Stage 9: Send Mask
Stage 3: Send pkey
Key Production System
Stage 1: Execute Key setup technique. Generate Pkey, Mkey

**Figure 5 Model of our proposed Ciphertext-Policy-ABE technique.**

Previous versions of Ciphertext-Policy ABE have presented access policies alongside ciphertexts in plaintext form, potentially compromising user attributes and privacy. In the existing CP-ABE scheme, verifying data owner authentication and ensuring outsourced data integrity are challenging. To address these issues, we propose a novel approach leveraging the BLS short signature scheme to authenticate data owners and verify data integrity, thereby enhancing protection against insider data theft. To further enhance access policy privacy, we incorporate a policy anonymization scheme into our proposed system. Listing-1 outlines the anonymization process, which employs hashing techniques for enhanced privacy protection.

**Listing-1:** Anonymization of policies for access

```
input: Policy for Access (PA)
output: Anonymized Policy for Access
Anonymizations(PA)
Procedure Anonymize(PA):
Parse the access policy.
Extract attributes not belonging to {AND, OR, relational_operators, +, =, gates threshold}.
for each attribute att:
    att = hashing(att)
end for
return PA
```

In an ABE technique, all insiders are considered legitimate users for accessing the genuine message. To prevent insider theft, a short signature strategy is employed, ensuring data owner authentication and shared data integrity. The access policy, represented as an access structure in Figure 6, is embedded into the ciphertext for access control. Utilizing Listing-1, access policies are anonymized. Before operations, InMessage undergoes encryption using the public key. Encryption and signature strategies are outlined in Listing-2, employing a set of leaf nodes (LN).

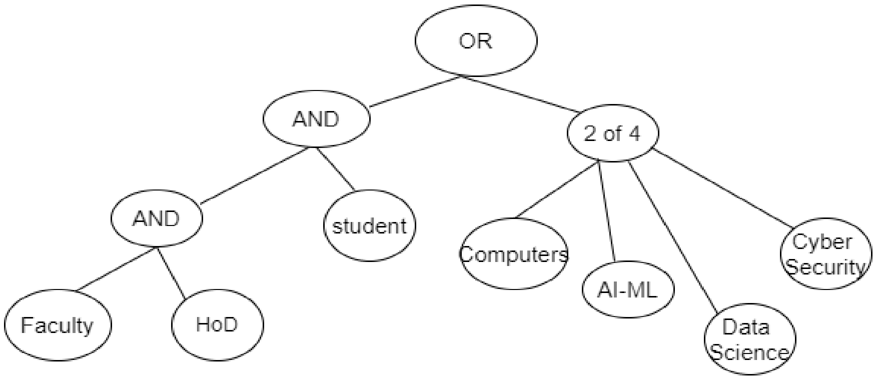

**Figure 6 Access Structure Illustration**

**Listing-2:** Encryption of Information

```
Procedure Sign Encipher (P Key, In Message, PA)
Anonymizations(PA)
if t node == t root then
while tree root node TRN do
    QTRN (0) = group of access structures AS
End while
```

```
end if
SC' = In Message. e (x, x) ψ AS; // ψ: random exponent, x: generators
SC = (hashing)AS
if tnode == leaf node then
while entire leaf nodes l node ∈ LN do
SCl = x Ql (0)
SC'l = H(attr(l)) Ql (0)
end while
end if
Signs (In Message, y)
```

The outcome results of the procedure in Listing-2 yield the Cipher text (CT), which can be represented as:

{Anonymizations(PA), Signs = hashing (In Message) y, SC= In Message. E (x, x) ψ AS; SC = (hashing)AS [SCl = xQl(0); SCl = H(attr(l))Ql(0) for all leaf nodes belonging to LN]}.

The signing process of signatures is implemented using the Boneh–Lynn–Shacham (BLS) method.[6] Assuming x is the generator of gap groups G1 along with the finite prime ordering of PR and a hash operation (hashing), shorter signatures can be computed and associated with the ciphertext to be shared. The structure of the data in the cloud will have three fields: a unique identity for the shared data, a signature, and an encrypted document along with the access policy, which is hidden.

Decryption is successful when the attributes of the access policy embedded within the ciphertext match the client's attributes in the cloud. If they do not match, the cloud client cannot perform decryption of the ciphertext. The shorter signature of BLS is utilized to verify the legitimacy of the information owner, thus preventing insider attacks. The techniques outlined in Listing-3 illustrate the decryption operation of our proposed methodology. The validate () function is used for the verification of signatures.

**Listing-3:** Decrypting Operation and Validation

```
Procedure Decrypt Validate (P Key, S Key u, Cipher text, short Signature, p key)
Decrypt Node (Cipher text, S Key, l)
if policy fulfilled by AS then
AS = Decrypt Node (Cipher text, S Key, In Message) = e(x, x) MAS
// Message M, Access Structure AS
SC' = InMessage. e (x, x) ψAS;
e (SC, D) = e(xψAS, x(ψ+M/ϒ)) // ψ and ϒ are randomly chosen exponents
In Message = SC' / (e (SC, D) / AS)
endif
validate (short Signature, p key)
```

## IV. EXPERIMENTATION AND RESULTS

The experimentation is done by Considering all tasks of the proposed methodology, by making use of standard computing systems with respective operating systems and mobile as IoT device. The Ciphertext-Policy ABE toolkit based on java [22][23] along with the library jPBC [24] are made use during the implementation of proposed operations. The quantity of attributes of clients are chosen in range of 20 to 200 in different intervals [25]. The results are evaluated for the performance of proposed works by comparing with various existing works proposed by researchers in [26][27][28][29][30][31]. The proposed strategies utilize anonymization of policies, further develops the security strategy and signature confirmation of the information owner and distinguishes insider's attacks. To accomplish this anonymizations of policies, SHA1-Secure Hashing Algorithm was utilized. Be that as it may, mentioned hashing technique presents an irrelevant overhead at the part of clients.

As demonstrated in Figure 7, existing Cipher text-Policy ABE techniques by hiding policies, the [26], [27], [28], [29], [30], [31], [32], [33] taken 0.107, 0.200, 0.260, 0.250, 0.210 and 0.130 seconds to produce a key of 20 attributes, wherein our proposed technique consumed only 0.100 seconds. At the same time, considering 200 quantity of attributes, key production times taken are 1.090, 1.584, 1.680, 1.640, 1.590 and 1.140 seconds, wherein the proposed technique, consumed 1.040 seconds, that is lower in comparison with the existing techniques considered for experimentation.

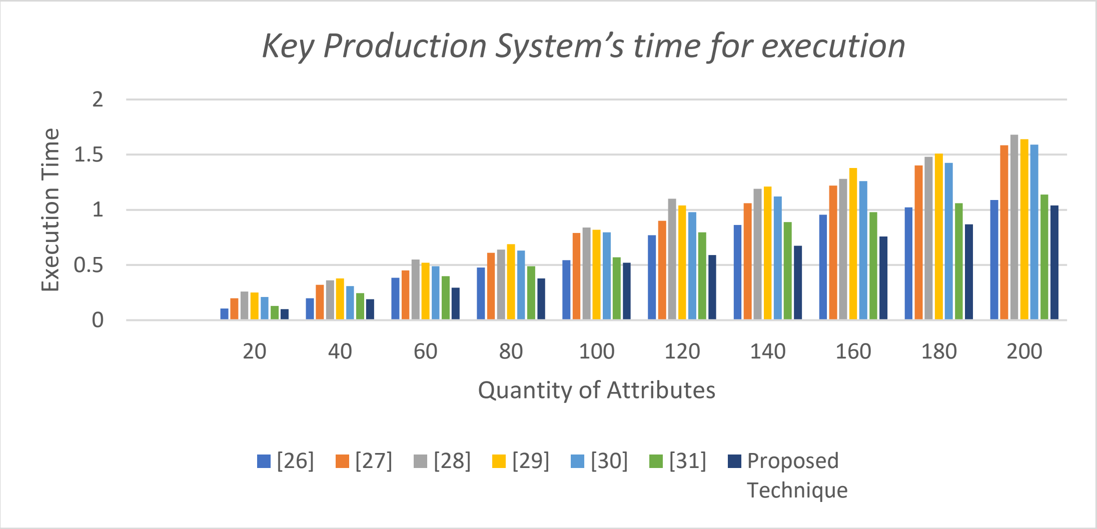

**Figure 7 Execution time of key production system**.

The comparisons of time utilized for encryption by the information owner is demonstrated in Figure 8. The methods existing considered for comparisons taken 0.490, 0.510, 0.563, 0.521, 0.543 and 0.520 seconds accordingly for 20 quantities of attributes, wherein the proposed technique consumed 0.485 seconds. At the same time, considering quantity of attributes as 200,

The time taken by the existing techniques recorded as 3.064, 3.300, 3.500, 3.460, 3.480 and 3.082 seconds respectively, that of proposed technique is 2.992 seconds, that is much lower time value than existing techniques.

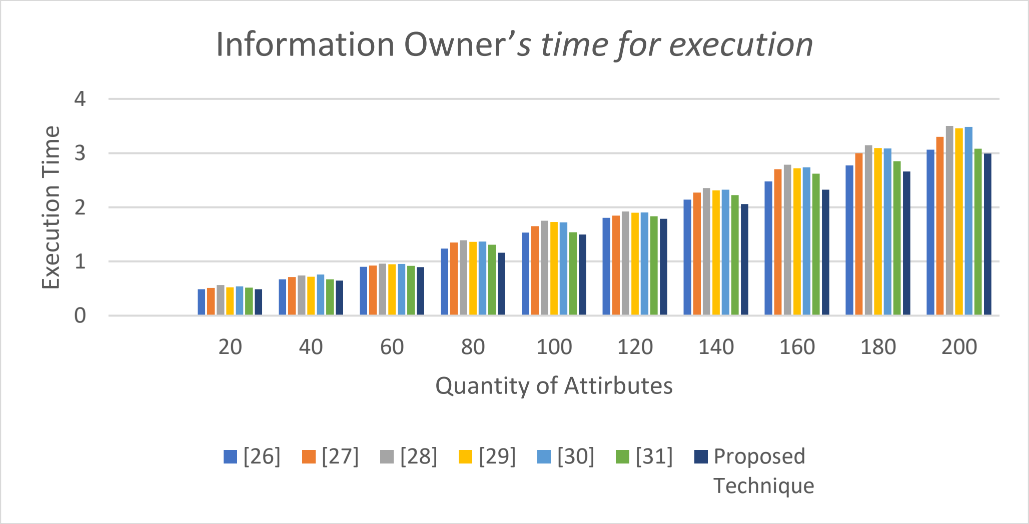

**Figure 8 Execution time of Information Owner.**

The technique proposed make use of hidden policies for accessing and experimentation proved that the approaches used deliver a comparatively better privacy and securities that already available approaches which are considered in this work. Recorded the time taken by the existing and proposed time for decryptions at client's end considering different value of quantity of attributes ranging from 20 to 200. The Figure 9 demonstrates the analysis of the same. Because of the use of the anonymization of the policies in the proposed approach, the time consumed by information clients appears little more than the approaches which are existing. At the same time, the approach proposed distinct methodology to secure from attacks by making use of scheme shorter signatures.

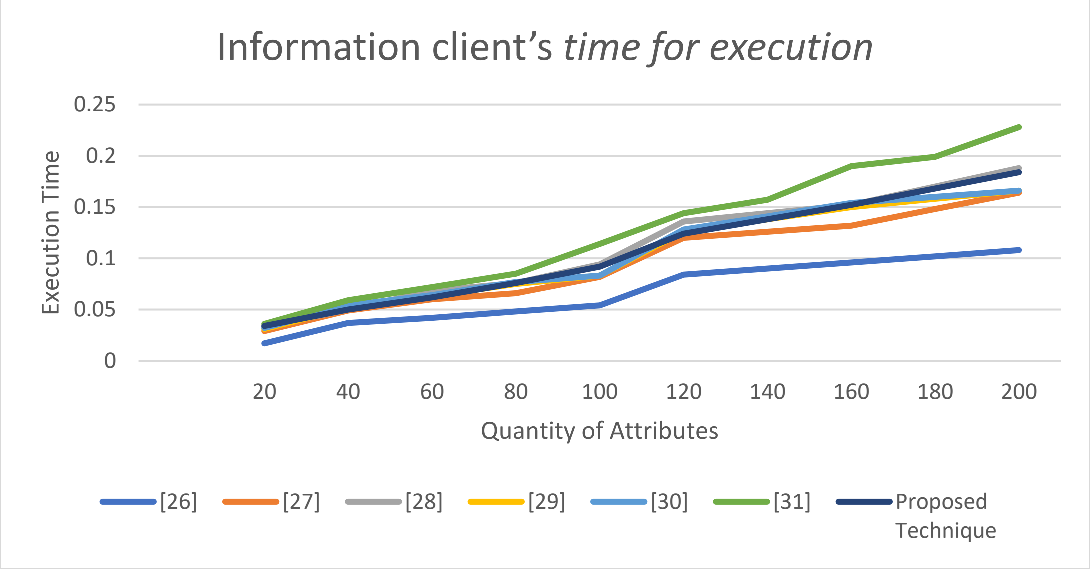

**Figure 9** Execution time of Information Client.

## V. DISCUSSION

The experimentation results affirm the efficiency of our proposed methodology in advancing the efficiency and security of Ciphertext-Policy Attribute-Based Encryption systems, particularly in IoT contexts. By significantly reducing key production and encryption times, our approach offers tangible benefits for real-world applications where speed and scalability are paramount. The findings hold significant implications for the design and implementation of IoT systems, promising faster and more secure data access while mitigating insider attacks and ensuring data integrity [34, 35].

Looking ahead, future research should continue to explore avenues for further optimization and resilience of our proposed methodology. Continued efforts to refine cryptographic operations and explore advanced anonymization techniques hold the potential to yield even greater improvements in performance and security. Also, assessing the applicability of our approach across diverse IoT scenarios and evaluating its resilience against emerging threats will be crucial for its widespread adoption and long-term viability in IoT ecosystems.

## VI. CONCLUSION

Utilizing attribute-based encryption techniques for encrypting information in cloud computing ensures security for outsourced data along with fine-grained access control. This paper introduces a novel scheme aimed at enhancing data privacy in the cloud. The proposed mechanism not only safeguards data in the cloud but also verifies data integrity against potential attacks. Through detailed discussion in this article, the methodology is outlined, and experimental results are presented, comparing them with six similar existing techniques. The comparative analysis demonstrates that the presented approach surpasses the performance of the six considered methods.

## REFERENCES

[1] Riad, K.; Hamza, R.; Yan, H. Sensitive and Energetic IoT Access Control for Managing Cloud Electronic Health Records. IEEE Access 2019, 7, 86384–86393.

[2] Wang, G.; Liu, Q.; Wu, J. Achieving fine-grained access control for secure sharing on cloud servers. Concurr. Comput. Pract. Exp. 2011, 23, 1443–1464.

[3] Zhu, H.; Wang, L.; Ahmad, H.; Niu, X. Key-Policy Attribute-Based Encryption with Equality Test in Cloud Computing. IEEE Access 2017, 5, 20428–20439.

[4] The Boneh-Lynn-Shancham Signature. Available online: https://en.wikipedia.org/wiki/Boneh_Lynn_Shacham (accessed on 25 December 2021).

[5] Sahai, A.; Waters, B. Fuzzy Identity-Based Encryption. In Proceedings of the Annual International Conference on the Theory and Applications of Cryptographic Techniques; Springer: Berlin, Germany, 2005; pp. 457–473.

[6] Miltiadis, M.; Virvilis, N.; Gritzalis, D. The insider threat in cloud computing. In International Workshop on Critical Information Infrastructures Security; Springer: Berlin/Heidelberg, Germany, 2011; pp. 93–103.

[7] R. K. Kalluri and C. V. G. Rao, ``Addressing the security, privacy and trust challenges of cloud computing," Int. J. Comput. Sci. Inf. Technol., vol. 5, no. 5, pp. 6094_6097, 2014.

[8] P. G. Shynu and K. J. Singh, ``A comprehensive survey and analysis on access control schemes in cloud environment," Inf. Technol., vol. 16, no. 1, pp. 19_38, 2016.

[9] R. K. Aluvalu and L. Muddana, ``A survey on access control models in cloud computing," in Proc. 49th Annu. Conv. Comput. Soc. India (CSI), vol. 1, 2015, pp. 653_664.

[10] Pan Jun Sun, Privacy Protection and Data Security in Cloud Computing: A Survey, Challenges, and Solutions, IEEE Access,2019, doi: 0.1109/ACCESS.2019.2946185

[11] M. S. Inamdar and A. Tekeoglu, ``Security analysis of open-source network access control in virtual networks," in Proc. 32nd Int. Conf. Adv. Inf. Netw. Appl. Workshops, May 2018, pp. 475_480.

[12] H. Takabi, ``Privacy aware access control for data sharing in cloud computing environments," in Proc. 2nd Int. Workshop Secur. Cloud Comput., 2014, pp. 27_34.

[13] X. Li, B. Yang, and M. Zhang, ``New construction of fuzzy identity-based encryption," in Proc. WASE Int. Conf. Inf. Eng., 2009, pp. 647_651.

[14] V. Goyal, O. Pandey, A. Sahai, and B. Waters, ``Attribute-based encryption for fine-grained access control of encrypted data," in Proc. 13th ACM Conf. Comput. Commun. Secur., 2006, pp. 89_98.

[15] S. D. C. di Vimercati, S. Foresti, R. Moretti, S. Paraboschi, G. Pelosi, and P. Samarati, ``A dynamic tree-based data structure for access privacy in the cloud," in Proc. IEEE 8th Int. Conf. Cloud Comput. Technol. Sci., Dec. 2016, pp. 391_398.

[16] L. Xue, Y. Yu, Y. Li, M. H. Au, X. Du, and B. Yang, ``Efficient attribute-based encryption with attribute revocation for assured data deletion," Inf. Sci., vol. 479, pp. 640_650, Apr. 2019.

[17] M. Basthikodi, W. Ahmed, Parallel algorithm performance analysis using OpenMP for multicore machines. Int. J. Adv. Comput. Technol. (IJACT) 4(5), 28–32 (2015)

[18] Mustafa Basthikodi and Waseem Ahmed, "Classifying a program code for parallel computing against hpcc", International Conference on Parallel Distributed and Grid Computing (PDGC), pp. 512-516, 2016.

[19] J. Wei, W. Liu, and X. Hu, ``Secure and efficient attribute-based access control for multiauthority cloud storage," IEEE Syst. J., vol. 12, no. 2, pp. 1731_1742, Jun. 2018.

[20] Shanthakumar et al, System for Fusion of Face and Speech Modalities Using DTCWT+QFT and MFCC+RASTA Techniques, Indian Journal of Science and Technology 2021;14(42):3144–3156. Doi: 10.17485/ijst/v14i42.1316

[21] C. K. M and M. Basthikodi, "Machine Learning Approaches for Abandoned Luggage Detection," 2023 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Mangalore, India, 2023, pp. 8-12, doi: 10.1109/DISCOVER58830.2023.10316663.

[22] D. Vaduganathan, ``Secure data sharing using attribute-based encryption with revocation in cloud computing," South Asian J. Eng. Technol., vol. 2, no. 15, pp. 145_150, 2016.

[23] Wang, J. Java Realization for Ciphertext-Policy Attribute-Based Encryption. 2012. Available online: https://junwei.co/cpabe/ (accessed on 10 December 2021).

[24] Ciphertext-Policy Attribute Based Encryption Toolkit. 2018. Available online: http://hms.isi.jhu.edu/acsc/cpabe/ (accessed on 10 December 2021).

[25] The Pairing-Based Cryptography Library. 2012. Available online: https://crypto.stanford.edu/pbc/ (accessed on 10 October 2021). Available online: https://health.data.ny.gov/api/views/tsg2-5hds/files/5ded175f-ecf3-4dd2-bb38-df464b137958?filename= NYSDOH_HospitalInpatientDischarges_SPARCS_De-Identified_2016.zip (accessed on 25 November 2021).

[26] Zhuo, S., Hong, Y. Y., & Palaoag, T. D. (2022, December 14). An Intelligent Cyber Security Detection and Response Platform. *International Journal for Research in Advanced Computer Science and Engineering*, *8*(12), 1–10. https://doi.org/10.53555/cse.v8i12.2167

[27] Helil, N.; Rahman, K. CP-ABE Access Control Scheme for Sensitive Data Set Constraint with Hidden Access Policy, and Constraint Policy. Secure Communications Networks 2017, 2017, 2713595.

[28] Sabitha, S.; Rajasree, M.S. Access control-based privacy preserving secure data sharing with hidden access policies in cloud. J. Syst. Archit. 2017, 75, 50–58.

[29] Zhang, Y.; Zheng, D.; Deng, R.H. Security and Privacy in Smart Health: Efficient Policy-Hiding Attribute-Based Access Control, IEEE Internet Things J. 2018, 5, 2130–2145.

[30] Wu, A.; Zheng, D.; Zhang, Y.; Yang, M. Hidden Policy Attribute-Based Data Sharing with Direct Revocation and Keyword Search in Cloud Computing. Sensors 2018, 18, 2158.

[31] Manoj Kumar, Arnav Kumar, Abhishek Singh, Ankit Kumar. Analysis of Automated Text Generation Using Deep Learning. International Journal for Research in Advanced Computer Science And Engineering; 7(4): 1-8.

[32] Odelu, V.; Das, A.K.; Khan, M.K.; Choo, K.R.; Jo, M. Expressive CP-ABE Scheme for Mobile Devices in IoT Satisfying Constant-Size Keys and Ciphertexts. IEEE Access 2017, 5, 3273–3283.

[33] Sachin Patil and D.V. Patil and Rahul Nejkar, Data Integrity Check in Cloud using Dispersal Code, International Journal of Engineering Research and Technology, Vol 1, 2012/11

[34] Sachin Patil and Sagar Gurav, Concealing Access Policies of Users from Clouds in Decentralized Access Control, Inventi Rapid: Cloud Computing, 2016/7/23

[35] P. Chinnasamy, P. Deepalakshmi, Ashit Kumar Dutta, Jinsang You and Gyanendra Prasad Joshi, Ciphertext-Policy Attribute-Based Encryption for Cloud Storage: Toward Data Privacy and Authentication in AI-Enabled IoT System, Mathematics 2022, 10, 68. https://doi.org/10.3390/math10010068.