International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Hybrid Cipher-Text Key Policy Attribute-Based Encryption (HCKP-ABE): The Performance Analysis and Scalability in Virtual Machines

Avuthu Siva Swaroopa Rani[1], Kondepati Rohini[2], Kilarapu Pavani[3], Jangala Rani Sai Sree[4], T Pavan Kumar[5], Pachipala Yellamma[6*]

Submitted: 11/01/2024 **Revised**: 17/02/2024 **Accepted**: 25/02/2024

Abstract: Attribute-based encryption (ABE) has appeared as a promising advanced cryptographic approach that has emerged is a viable option for secure sharing of data. ABE is particularly useful in scenarios where access to the data should be limited to authorized users depending on their attributes. AB encryption, such as Ciphertext Policy, enables data owners to specify fine-grained access control policies for encryption, based on user attributes. This assures that only people with the necessary credentials may decode and access the encrypted data, providing an additional layer of security. Key Policy ensures secure data access by attaching rules to encrypted information, enabling decryption only for users with matching characteristics. However, in most previous schemes, the execution time for Cipher-Text Key policy Attribute-Based Encryption has been a significant challenge. To overcome this issue, the proposed works have implemented an effective encryption and decryption process. In this paper provide extensive research of the intricacies of the implementation process, evaluating the performance of the developed Hybrid Cipher-Text Key policy Attribute-Based Encryption (HCKP-ABE) system and presenting a comparative analysis of existing schemes. Hybrid Cipher-Text Key policy Attribute-Based Encryption offers significant performance advantages over existing schemes such as KP-ABE RSA & AES. Our performance evaluation revealed that the execution time increases with the number of users and attributes. The results show HCKP-ABE has the smallest key size (512 bits) and the fastest execution time (0.0189 seconds). KP-ABE has a key size (512 bits) and a longer execution time (0.0483 seconds). RSA & AES has the largest key size (2048 bits) and the slowest execution time (0.0295 seconds). HCKP-ABE's pairing-based cryptography enables efficient attribute-based encryption and decryption, and it is a suitable choice for scenarios requiring fine-grained access control and secure data sharing, particularly when the number of users is relatively small compared to the number of attributes.

Keywords: *Cloud, Cryptography, Decryption, HCKP Attribute-based Encryption, KP attribute-based Encryption, RSA.*

1. Introduction

In today's tech-driven landscape, cloud computing has become a go-to solution, providing easy access to a variety of services. This model, utilizing the internet for distributed resources, is reshaping how businesses manage their IT infrastructure. Cloud Computing platforms can quickly allocate computing power and resources according to the user's needs [1]. Privacy protection is a significant worry, revolving around the prevention of unauthorized access to sensitive data [2]. With multiple users utilizing services across a network, ensuring data privacy becomes a complex task. Service providers play a pivotal role in guaranteeing data confidentiality, with encryption emerging as a practical solution to secure data and limit access to authorized individuals [3]. Ensuring data integrity is equally important, particularly in collaborative settings where several individuals contribute to shared projects using cloud platforms [4]. The focus here is on maintaining data consistency during transmission and storage. While encryption is commonly employed, its time-consuming nature can pose vulnerabilities [5]. Authorized personnel with access to sensitive data may misuse it [6], requiring robust security measures andcontinuous monitoring of user activities within the cloud.

1.1. Cryptography

Cryptography is a fundamental solution. Often referred to as a secret code, in cryptography during transmission and storage, the data is converted into an unreadable format [7]. As organizations grapple with these security challenges in the cloud, cryptography emerges as a cornerstone strategy to keep data secure and ensure it reaches its intended destination. This transformational process is critical to protecting data secrecy by guaranteeing that only authorized personnel may understand and access the original data [8]. As the digital realm becomes increasingly interconnected and dependent on cloud-based services, the role of cryptography extends beyond mere confidentiality.

Among the various types of asymmetric key encryption, AB Encryption (ABE) stands out as a particularly relevant and versatile approach. ABE is designed to enable fine-grained encryption data access control,

[1,2,3,4,5,6] *Koneru Lakshamaiah Education Foundation, Guntur, India 522502*
ORCID ID : 0000-0002-6291-7418
** Corresponding Author Email: pachipala.yamuna@gmail.com*

allowing for access restrictions based on specific attributes or characteristics [8].

The working methodology of AB Encryption involves the following key aspects:

Attribute-Based Key Generation: Users are assigned a private key based on their attributes. These attributes could be, for example, roles, job titles, or any other defining characteristic.

Encryption: To encrypt data, an access policy that defines the qualities required for decryption is utilized. Only users with the appropriate private key quality and access policies can decode and view the data.

Fine-Grained Access Control: AB Encryption allows for precise control over who can access specific pieces of data [9]. This granularity is particularly valuable in scenarios where different users may need access to different subsets of information.

Revocation: ABE systems often include mechanisms for key revocation. If a user's attributes change or if access needs to be revoked, the corresponding private key can be updated accordingly.

Ciphertext-Policy Attribute-based Encryption:

CP-ABE is a way of keeping information safe that gives a lot of control over who gets to see what. In CP-AB Encryption, objects are encrypted with rules attached to them, kind of like saying, "Only people with a certain job title or characteristic can read this." Users have private keys with their own set of characteristics, and for them to see the encrypted information, their characteristics must match the rules attached to it [10]. It's often used in things like secure cloud storage and healthcare data sharing, making sure that only the right people with the right characteristics can unlock and access specific pieces of information.

Key-Policy Attribute-Based Encryption:

KP-ABE is a way of keeping information secure while giving a lot of control over who can access it. In KP-ABE, users have private keys based on their characteristics, just like in other types of encryptions. What makes KP-ABE different is that the rules about who can see the encrypted data are attached to the data itself. So, it's like saying, "Only people with specific characteristics can read this particular piece of information." When the data is encrypted, it comes with a set of rules, and users can only unlock and see it if their private key matches those rules [11]. This approach is handy when you need to manage who gets to see what in a dynamic way, especially when users might have changing sets of characteristics.

Our proposed method HCKP-ABE offers a greater degree of control over access policies compared to traditional ABE schemes. Enables a fine-grained and dynamic approach to data sharing. Accommodates diverse access scenarios by combining features of both KP-ABE and CP-ABE. Mitigates potential collusion attacks by distributing policy elements between ciphertext and keys. Suitable for scenarios where both data owners and users need to define access rules. Strengthens security measures by requiring a match between both attribute sets for decryption. Reduces the complexity of key management compared to separate KP-ABE and CP-ABE systems. Streamlines are the process of assigning and revoking access rights.

The paper's organization of the information follows: Section 2 provides a detailed assessment of existing research on hybrid encryption schemes. Section 3 discusses the present methods for safe data transfer utilizing hybrid cipher text key policy attribute-based encryption for protecting data in virtual machines. We go over the encryption procedure in depth. Section 4 gives the findings and comparative analysis of our proposed technique. Finally, Section 5 includes a full discussion of the study paper's findings and conclusions.

2. Literature Survey

The topic of attribute-based encryption (ABE) and its role in protecting patient data privacy in collaborative eHealth systems was explored by, Hung Kook Park, Jong Wook Kim, Beakcheol Jang, and Kennedy Edemacu in 2019. By removing geographical and accessibility constraints from healthcare service delivery, collaborative eHealth makes it easier for different service providers to gather and share patient data [12]. The security, revocation ability, and efficiency of the surveyed schemes are compared and evaluated.

A unique updatable CP-ABE was presented in 2019 by, Jing Xu, Baocang Wang, Yan Liu, and Zhenhua Liu to handle concerns about security pertaining to restricted access control in intricate commercial applications. Traitor revocation and white-box traceability are supported by the programie. It is determined that the suggested technique, which achieves proper revocation and ciphertext updates, is more efficient than earlier works [13]. The scheme's security is predicated on the Decisional q-Bilinear DDH assumption, while its traceability is dependent on the l-Strong Diffie-Hellman assumption. The suggested scheme's efficiency is demonstrated by the experimental findings.

2019 saw the introduction of Anonymous Attribute-Based Broadcast Encryption (A2B2E), a unique approach for safe data sharing in cloud computing by

Jianfei Sun, Hao Zhang, and Hu Xiong. With the help of this approach, which introduces a hidden access policy, data owners can safely communicate information with numerous participants within a preset receiver set while still following access regulations that have been stated. They also create an online/offline attribute-based encryption idea, [14] a verified outsourcing attribute-based encryption decryption technique, and an A2B2E scheme for safe data sharing.

A workable multi-keyword searchable encryption strategy was presented by X. Yao, J. Sun, L. Ren, and S. Wang in 2019 for safe data retrieval in cloud-based data sharing systems. [15] The proposed scheme enhances security against potentially untrustworthy cloud servers through attribute revocation via ciphertext updates, supports efficient multi-keyword searches, and effectively manages attribute changes by combining CP attribute-based encryption with evaluating concepts. According to substantial experimental results, the system shows resilience against keyword and selected plaintext assaults under the generic group model.

In order to overcome issues with safely sharing data, S. K. Mohanty and R. Ahuja (2020) suggested a scalable AB access control approach for cloud storage. Their system, in contrast to existing solutions, provides scalability by expanding attribute-based encryption (ABE) ciphertext policy. The main innovation is the hierarchical user structure that allows for the flexible delegation of access privileges [16]. Through implementation, the authors show the scalability and efficiency of their method and explicitly prove its security on the basis of ciphertext-policy ABE's security.

A Lightweight Revocable Hierarchical Attribute-Based Encryption (LW-RHABE) method was presented by Ximeng Liu, Mohammad-Reza Sadeghi, and Mohammad Ali in 2020 and was designed specifically for cloud-based Internet of Things (IoT) networks. [17] The plan seeks to remedy the shortcomings of the current Attribute-Based Encryption (ABE) systems, including the inability to efficiently encrypt and decode data as well as the restrictions on user revocation and flexible key delegation. LW RHABE gives user computational efficiency first priority, with the cloud server handling most computing tasks.

In 2020 J. L. G. Compean, M. Morales-Sandoval, M. H. Cabello, and H. M. Marin-Castro, "Attribute-Based Encryption Approach for Storage, Sharing and Retrieval of Encrypted Data in the Cloud," propose a comprehensive security methodology, named FABECS, for cloud storage addressing concerns related to data encryption, sharing, and retrieval. The approach is based on ab encryption, offering methods for managing access to encrypted data and information retrieval via search access control. [18] The authors prove the accuracy and performance of FABECS for sharing the document, archiving, and extraction in a cloud-based setting by rigorous analysis and testing.

The below Table 1 provides a literature review of existing methodologies for enhancing security. It provides a comparison of different approaches, including hybrid cryptographic algorithms, RSA, AES, CP-ABE, KP-ABE, ABE, elliptic curve cryptography, and more.

Table – 1: Literature Survey of Existing works

S. No	Author	Title	Methodology	Drawbacks
1.	D. Tiwari, A. Singh, A. Prabhakar	Performance Analysis of AES, RSA and Hashing Algorithm Using Web Technology [19]	RSA	The research primarily focuses on encryption and decryption time as performance parameters, overlooking crucial considerations such as memory utilization, and key generation time, neglecting a comprehensive evaluation of various performance factors.
2.	G. Dini, S. Yu, M. Rasori, and P. Perazzo	Indirect Revocable KP-ABE With Revocation Undoing Resistance [20]	KP-ABE	The new ABE scheme has slightly higher computational costs than previous schemes.
3.	H. T. Elshoush, M. A. Elsadig, S. Hasan, B. Barry, S.	A Polymorphic Advanced Encryption Standard	Polymorphic(PAES).	P-AES is a fixed algorithm, which means that it cannot be easily modified to adapt to new threats or

	Naserelden, and A. Altigani	– A Novel Approach. [21]		vulnerabilities.
4.	Mohammad Rafeek Khan, Kamal Upreti, Mohammad Imran Alam, Haneef Khan, Shams Tabrez Siddiqui, Mustafizul Haque and Jyoti Parashar	Analysis of Elliptic Curve Cryptography & RSA. [22]	RSA and ECC	ECC is more complex than RSA, and it can be more difficult to implement correctly. This can make ECC more susceptible to implementation errors, which could also introduce security vulnerabilities
5.	A. Sajid, S. M. Mohsin, M. Awais, M. B. Qureshi, I. Mustafa, I. U. Khan, S. Aslam	A Lightweight Post-Quantum Lattice-Based RSA for Secure Communications [23]	Post-Quantum Lattice-Based RSA	The algorithm's performance is inferior to that of the RSA algorithm. This is a result of the algorithm's substitution of lattices for prime integers.
6.	Hong B, Chen J., Zhang K, and Qian H.	Multi-Authority Non-Monotonic KP-ABE With Cryptographic Reverse Firewall [24]	KP-ABE	The scheme is computationally expensive, as it requires multiple authorities to cooperate and is vulnerable to collusion attacks, as multiple authorities could collude to decrypt a ciphertext.
7.	Li, W., Jin, Z., Zhang, H., and Q. Wen	An Efficient ABE Scheme with Verifiable Outsourced Encryption and Decryption [25]	ABE	The scheme is vulnerable to attacks from malicious servers, as the scheme relies on untrusted servers to perform encryption and decryption.
8.	N. Guo, Y. Zhang, J. Li, and N. Chen	Efficient CP-ABE Scheme With Shared Decryption in Cloud Storage [26]	CP-ABE-SD	The scheme may be vulnerable to attacks from malicious users, as the scheme relies on multiple users to decrypt the data.
9.	S. Namasudra and S. Das	Multiauthority CP-ABE-based Access Control Model for IoT-enabled Healthcare Infrastructure [27]	CP-ABE with ECC	It is vulnerable to attacks from malicious attribute authorities, as several attribute authorities are used by the programme to minimise work cost.
10.	J. Li, H. Yan, and Y. Zhang	Constant Size Ciphertext Distributed CP-ABE Scheme With Privacy Protection and Fully Hiding Access Structure [28]	CP-ABE	The scheme suffers from efficiency drawbacks due to each attribute's duplicate for each location in the access tree. This can lead to significant overhead during encryption and decryption, especially for large access structures.

3. Methodology

Setup Algorithm: The algorithm Setup establishes cryptography parameters and also generates master and public keys. The security of the Ciphertext Policy-ABE system relies on the difficulty of certain mathematical problems associated with cryptography, such as the discrete logarithm problem. The setup complexity is

typically constant, as it involves generating keys and configuring cryptographic parameters.

Key Generation Algorithm: The master key is used by the Key-Generation algorithm to process user-specific private keys depending on their properties. The system's security stems from the challenge of deriving a user's private key without the master key. The key generation complexity depends on the attributes associated with a user and the chosen cryptographic primitives. In terms of the quantity of qualities, it is frequently linear.

Encrypt Algorithm: The Encrypt algorithm utilizes elliptic curve operations and bilinear pairings to create ciphertext. Security is grounded in the presumed difficulty of deriving the original message without the required attributes. Encryption complexity is generally affected by the complexity of pairing operations and the structure of the access structure. It can vary based on the chosen pairing function.

Decrypt Algorithm: The Decrypt algorithm determines whether the user's characteristics match the ciphertext's access structure. The level of security depends on how tough the selective decrypting is without the proper attributes. The quantity of attributes and access structure determine the decryption complexity. It often involves pairing operations and elliptic curve point operations.

Performance Metrics: Performance metrics include average execution time and total execution time. These metrics help evaluate the efficiency of the HCKP-ABE system in actual life scenarios. The measurement of execution times depends on factors such as the amount of data, attributes, and recurrences of users. The obstacle is influenced by the underlying cryptographic operations and system-specific configurations.

Experimental Design: The experimental design involves systematically varying the number of users, attributes, and repetitions. This approach aims to understand the impact of these parameters on system performance. The challenge of experimental design is influenced by the number of experiments conducted and the need for randomization to avoid bias.

Statistical Analysis: Statistical analysis, such as ANOVA, assesses the significance of observed differences in execution times. It provides insights into the impact of varying parameters on system performance. Statistical analysis complexity depends on the chosen tests and the size of the dataset.

Results Presentation: Presenting results involves creating visualizations and tables to communicate the relationships between parameters and execution times effectively. The complicated pattern of the presentation of the results based on the selected visualization methods and the need for clarity in conveying the findings.

Discussion: The discussion interprets the results in view of the theoretical foundations of Ciphertext Policy-ABE. It explores how variations in parameters impact system efficiency and security. The complexity lies in providing meaningful interpretations and identifying potential limitations of the study.

The above theoretical analysis emphasizes the reliance on the theoretical foundations of cryptography and HCKP-ABE, with complexities varying based on the specific cryptographic operations and experimental design considerations.

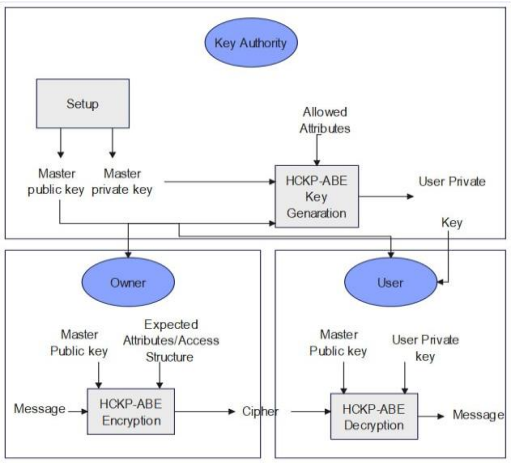

Fig. 1. Proposed Architecture of HCKP-ABE

The proposed Architecture as shown in Figure 1 is a kind of public-key encryption known as hybrid ciphertext-policy attribute-based encryption (HCKP-ABE) blends the ideas of key-policy attribute-based encryption (KP-ABE) and ciphertext-policy attribute-based encryption (CP-ABE). The ciphertext and the access policy are linked in CP-ABE, whereas the encryption key and the access policy are linked in KP-ABE. By linking the access policy to the encryption key and the ciphertext, HCKP-ABE combines these two methods. This makes it possible to define and administer access controls with greater flexibility.

The implementation of ABE using the HCKP-ABE involved a systematic and meticulous approach, encompassing several stages:

3.1. Mathematical Representation

- U be the user count.
- A be the amount of attributes
- T_{avg} be the average execution time in seconds,
- T_{total} be the overall execution time in seconds

Number of Users:

• *U* represent the number of Users in HCKP-ABE system

Number of Attributes:

• *A* represents the number of attributes associated with users in your HCKP-ABE system.

Average Execution Time:

• T_{avg} represents the average execution time for a particular operation (e.g., encryption, decryption) measured in seconds.

• $T_{avg} = \frac{T_{total}}{u \times number\ of\ repetitions}$

Total Execution Time:

• $Ttotal$ represents the total execution time for a specific operation measured in seconds. It is the sum of execution times over all repetitions.

• $T_{total} = U \times number\ of\ repetitons \times T_{avg}$

3.2. Algorithm: HCKP-ABE

- *Setup:* (PK, MK) ←Setup ()

The Setup process creates the public key (PK) and master key (MK). It typically involves selecting cryptographic parameters and creating the necessary components for the encryption and decryption processes.

- *Encrypt:* CT←Encrypt(PK,M,A)

The public key (PK), a message (M), and an access structure (A) are all inputs to the Encrypt algorithm. It generates the encrypted text (CT). The characteristics needed to decipher the ciphertext are specified in the access structure A.

- *Key Generation:* SK←Key Generation (MK,A[])

A user's private key (SK) is created using the Key Generation algorithm using the user's attributes ((A[])) and MK.

- *Decrypt:* M←Decrypt(PK,CT,SK)D

The Decrypt algorithm accepts as inputs the public key (PK), the ciphertext (CT), and the user's private key (SK). It returns the decrypted message if the user's characteristics match the access structure given in the ciphertext.

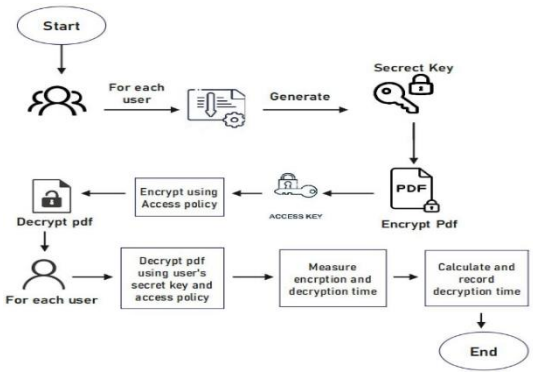

Fig. 2. Workflow of HCKP-ABE

The above Figure 2 depicts the process of generating a secret key and access policy for a HCKP-ABE system, measuring its performance, and recording the performance metrics. The process begins with defining the number of users and attributes. Next, secret keys are generated for each user, and access policies are generated for each attribute. The PDF is then encrypted using the generated keys and policies. Encryption and decryption times are measured to assess the system's performance. The total execution time, including decryption, encryption, setup, and key creation, is calculated. Finally, performance metrics, including the number of users, attributes, setup time, time spent on key creation, decryption, encryption, and overall execution are all tracked for analysis.

Algorithm: Proposed Algorithm of HCKP-ABE

Input:

U_{max}: Maximum number of users

A_{max}: Maximum number of attributes

R_{max}: Maximum number of repetitions

file_size: Size of the file to be encrypted (e.g., 5MB)

access_structure: An access structure with the properties specified that are needed to unlock the file.

Output:

performance_metrics: A list of performance metrics for each combination of parameters.

Initialize performance metrics as an empty list

for *U* in range (1, U_{max} + 1)

  for *A* in range (1, A_{max}+ 1)

    for *R* in range (1, R_{max}+ 1)

      # Setup HCKP-ABE system

      $T_{start} = T$

      M_{pk}, M_{sk} = setup hckp-abe (*U*, *A*)

      $T_{end} = T$

$T_{setup} = T_{end} - T_{start}$ ---- (1)

Create users and generate their keys

$K_{gen(start)} = T$

users = create users(U)

initializes an empty list named keys

for user in users:

generating user keys (K_{gen} cp abe (user, M_{pk}, M_{sk}))

$K_{gen(end)} = T$

K_{gen} (time) = $K_{gen(end)} - K_{gen(start)}$ ---(2)

Encrypt file using different access structures

$T_{encrypt}$ (start) = T

For access structure in generate access structures (A)

$C = E_{hckp\text{-}abe}$ (file size, access structure, M_{pk})

$T_{encrypt}$ (end) = T

$T_{encrypt} = T_{encrypt}$ (End)- $T_{encrypt}$ (start) ---(3)

Decrypt for each user and measure execution time

$T_{decrypt}$ (start) = T

$T_{decrypt} = 0$

for user in users

for key in keys

$T_{decrypt} = D_{cp\text{-}abe}$ (C, key)

$T_{0(decrypt)} += T_{decrypt}$

$T_{decrypt}$ (end) = T

$T_{decrypt} = T_{0(decrypt)}$ / (U* length of keys)

Calculate total execution time

$T_{total} = T_{setup} + K_{gen}$ (time) + $T_{encrypt} + T_{decrypt}$ ---(4)

#Calculate the total number of executions:

total Executions = $U * R$

#Calculate the average execution time:

$T_{avg} = T_{total}$ / total executions ---(5)

Record Performance Metrics ((U, A, T_{setup}, K_{gen} (time), $T_{encrypt}$, $T_{decrypt}$, T_{total}, T_{avg}))

Display or save performance metrics

Table – 2: Notations and Terminologies

Notations	Description
U	*Number of Users*
A	*Number of Attributes*
U_{max}	*Maximum number of Users*
A_{max}	*Maximum number of attributes*
T	*Time*
T_{total}	*Total execution time*
T_{setup}	*Setup time*
T_{start}	*Setup start time*
T_{end}	*Setup end time*
M_{pk}	*Master public key*
M_{sk}	*Master Secret key*
K_{gen}	*Key generation*
$K_{gen(start)}$	*Key generation start time*
$K_{gen(end)}$	*Key generation end time*
$T_{encrypt}$	*Encryption time*
$E_{cp\text{-}abe}$	*Encrypt CP-ABE*
$T_{decrypt}$	*Decryption time*
$T_{0(decrypt)}$	*Total decryption time*
C	*Ciphertext*
T_{aveg}	*Average execution time*
R	*Number of Repetitions*

4. Result Analysis

The Proposed models present the performance evaluation in the following section. In Ubuntu, all programmes are written in the C programming language. Windows 11 with an Intel (R) Core (TM) i5-7400 CPU running at 3.00 GHz with 8.00 GB of RAM serve as the simulation platform. The outcomes of the implementation are displayed.

The key size analysis in Figure. 3 revealed that the secret key size increases exponentially as the no. of attributes ascends. This observation highlights a trade-off between efficiency and security in ABE implementations. Larger key sizes enhance security but also increase computational overhead.

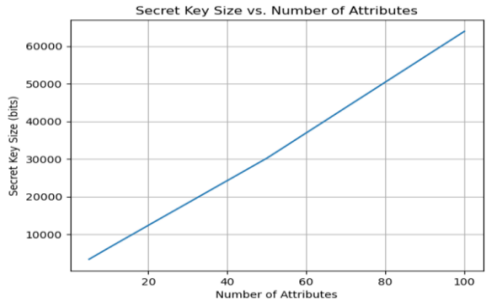

Fig. 3. Secret Key Vs Number of Attributes

Effect of Users:

As the Figure 4 shows that number of users increases, there is a consistent upward trend in average execution times for both encryption and decryption. This suggests that the algorithm's performance is directly impacted by the sheer number of users participating in the procedure. The increase in average execution time is not necessarily linear; however, there is a clear positive correlation.

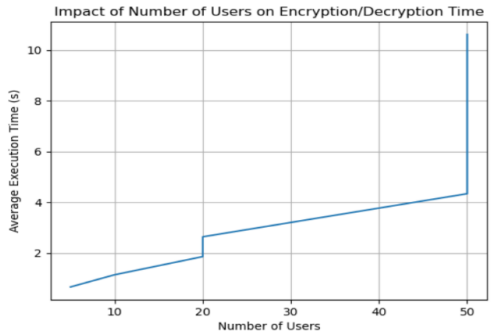

Fig. 4. Impact of Users

Effect of Attributes:

As shown in the figure 5, the impact of the number of attributes is more pronounced in certain scenarios. Comparing the cases where the number of attributes changes from 5 to 20, the increase in average execution time is more significant when the number of users is lower. For example, when moving from 5 users and 5 attributes to 5 users and 20 attributes, the average execution time increases noticeably. This is in contrast to the case of 20 users and 5 attributes to 20 users and 20 attributes, where the increase is still present but less pronounced.

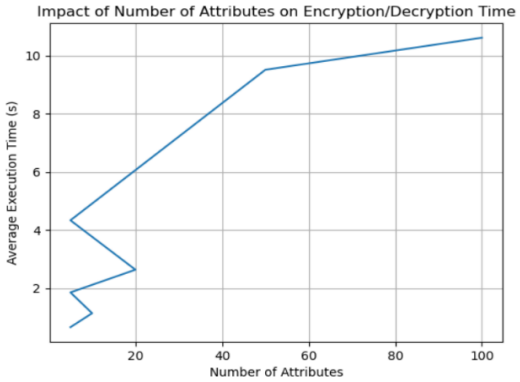

Fig. 5. Impact of Number of Attributes

In the Table – 3, we can observe the Average execution time and Total execution time for different number of Users and Attributes

Table – 3: Average Execution Time and Total Execution time

No. of Users	Attributes	Master Key Size (Bytes)	Avg. Execution Time (s)	Total Execution Time (s)
5	5	356	0.657	13.14
10	10	363	1.137	22.74
20	5	372	1.853	37.06
20	20	383	2.634	52.68
50	5	391	4.333	86.66
50	50	398	9.510	190.2
50	100	405	10.612	212.24

From Graph Figure 6, we can observe the clear pattern. As expected, the more users and attributes, the longer it will take to encrypt and decrypt the file. However, altering the quantity of attributes has no impact on the encryption's performance, nearly as much as increasing the number of users does. For example, with the case of 5 users and 50 attributes, the encryption and decryption process take roughly 30 seconds (1.469s on average) to finish. However, the inverse case of 50 users and 5 attributes per user takes more than double the time, taking almost 90 seconds (3.94s on average) to complete. Thus, it will be computationally less feasible to have 1k users than to have 1k attributes per user.

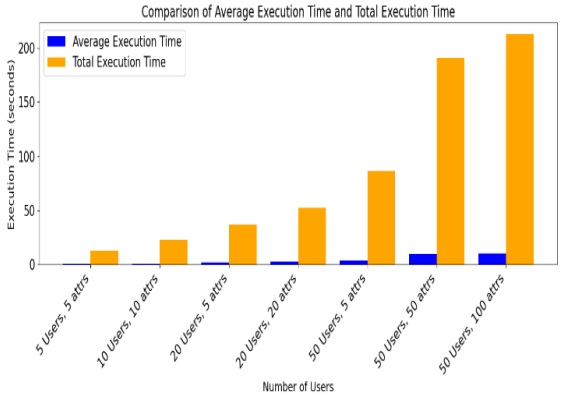

Fig. 6. Average Execution Time Vs Total Execution Time

The practical performance of our proposed HCKP-ABE through experimental evaluation on the data sets. Also, we compared KP-ABE, RSA & AES scheme for providing a performance comparison using our scheme. The experimental results demonstrate that our model is appreciable performance advantages on the aspect of total execution time compared with other schemes.

Table – 4: Comparative Analysis of KP-ABE, RSA& AES and HCKP-ABE

S.no	Algorithms	Key Size(bits)	Execution Time(sec)
1.	KP-ABE [20]	512	0.0383
2.	RSA & AES [19]	2048	0.118
3.	HCKP-ABE	512	0.0189

The aforementioned Table 4 shows that HCKP-ABE has the shortest key size (512 bits) and the quickest execution time (0.0189 seconds). The key size of KP-ABE is 512 bits, and its execution time is greater at 0.0483 seconds. Both AES and RSA have the slowest execution times (0.0295 seconds) and the highest key sizes (2048 bits). The findings show that, in comparison to different approaches, our suggested solution executes faster.

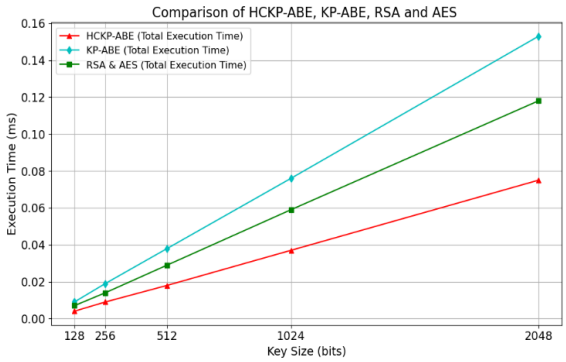

Fig. 7. Comparison of Execution Time

From the above Graph Figure, demonstrates the superior total execution time performance of our HCKP-ABE compared to KP-ABE, RSA & AES. This advantage stems from HCKP-ABE's utilization of pairing-based cryptography, which enables efficient attribute-based encryption and decryption operations and apart from total execution time, other factors to consider when choosing an encryption scheme include security strength, flexibility, and scalability. HCKP-ABE offers a balance of these factors, making it a suitable choice for various applications, particularly those requiring efficient AB encryption and fine-grained access control.

5. Conclusion

In this paper, examine the HCKP-ABE's performance in relation to several factors, including the quantity of attributes, the number of users, and the encryption and decryption times. By using these parameters calculated the average execution time and total execution time of the proposed model. The proposed also performed a comparative analysis on the aspect of execution time with KP-ABE, RSA & AES. The results show HCKP-ABE has the smallest key size (512 bits) and the fastest execution time (0.0189 seconds). KP-ABE has a key size (512 bits) and a longer execution time (0.0483 seconds). RSA & AES has the largest key size (2048 bits) and the slowest execution time (0.0295 seconds). As proposed can see the experimental results, HCKP-ABE is more scalable than other types of public-key encryption which highlights the importance of considering both the users and attributes when designing access control policies, ensuring an optimal balance between security and attribute-based encryption, and offering a versatile and adaptive solution for organizations seeking effective access control in dynamic environments.

References

Acknowledgments

This research was supported/partially supported by [KL University, Guntur, India]. We thank our colleagues who provided insight and expertise that greatly assisted the research, of this paper.

Author contributions

Siva Swaroopa Rani Avuthu: Conceptualization,

Methodology, Software, Validation, Writing-Reviewing and

Editing **Rohini Kondepati:** Formal analysis, Investigation,

supervision **Pavani Kilarapu:** Data curation, Resources **Rani**

Sai Sree Jangala: Writing-Original draft preparation **Pavan**

Kumar T : Visualization **Yellamma Pachipala:** Software,

Validation.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Y. Yu, J. Shi, H. Li, Y. Li, X. Du and M. Guizani, "Key-Policy Attribute-Based Encryption With Keyword Search in Virtualized Environments", IEEE Journal on Selected Areas in Communications, vol. 38, no. 6, pp. 1242-1251, June 2020.

[2] L. Zhang, G. Hu, Y. Mu and F. Rezaeibagha, "Hidden Ciphertext Policy Attribute-Based Encryption With Fast Decryption for Personal Health Record System", IEEE Access, vol. 7, pp. 33202-33213, 2019.

[3] L. Zhang, Y. Cui and Y. Mu, "Improving Security and Privacy Attribute Based Data Sharing in Cloud Computing", IEEE Systems Journal, vol. 14, no. 1, pp. 387-397, March 2020.

[4] C. Ge, W. Susilo, J. Baek, Z. Liu, J. Xia and L. Fang, "Revocable Attribute-Based Encryption With Data Integrity in Clouds", IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 5, pp. 2864-2872, 1 Sept.-Oct. 2022.

[5] A. Chiquito, U. Bodin and O. Schelén, "Attribute-Based Approaches for Secure Data Sharing in Industrial Contexts", IEEE Access, vol. 11, pp. 10180-10195, 2023.

[6] Karimunnisa, S., Pachipala, Y. "Task Classification and Scheduling Using Enhanced Coot Optimization in Cloud Computing", In: (2023) International Journal of Intelligent Engineering and Systems, 16 (5), pp. 501-511.

[7] R. Imam, Q. M. Areeb, A. Alturki and F. Anwer, "Systematic and Critical Review of RSA Based Public Key Cryptographic Schemes: Past and Present Status", IEEE Access, vol. 9, pp. 155949-155976, 2021.

[8] K. Pavani, J. R. S. Sree, A. S. S. Rani, K. Rohini, T. P. Kumar and P. Yellamma, "Data Security and Privacy Issues in Cloud Environment", In: 2023 5th International Conference on Smart Systems and Inventive Technology (ICSSIT), Tirunelveli, India, 2023, pp. 788-793.

[9] J. Fu and N. Wang, "A Practical Attribute-Based Document Collection Hierarchical Encryption Scheme in Cloud Computing", IEEE Access, vol. 7, pp. 36218-36232, 2019.

[10] H. Li, K. Yu, B. Liu, C. Feng, Z. Qin and G. Srivastava, "An Efficient Ciphertext-Policy Weighted Attribute-Based Encryption for the Internet of Health Things", IEEE Journal of Biomedical and Health Informatics, vol. 26, no. 5, pp. 1949-1960, May 2022.

[11] H. Yin et al., "CP-ABSE: A Ciphertext-Policy Attribute-Based Searchable Encryption Scheme", IEEE Access, vol. 7, pp. 5682-5694, 2019.

[12] K. Edemacu, H. K. Park, B. Jang and J. W. Kim, "Privacy Provision in Collaborative Ehealth With Attribute-Based Encryption: Survey, Challenges and Future Directions", IEEE Access, vol. 7, pp. 89614-89636, 2019.

[13] Z. Liu, J. Xu, Y. Liu and B. Wang, "Updatable Ciphertext-Policy Attribute-Based Encryption Scheme With Traceability and Revocability", IEEE Access, vol. 7, pp. 66832-66844, 2019.

[14] H. Xiong, H. Zhang and J. Sun, "Attribute-Based Privacy-Preserving Data Sharing for Dynamic Groups in Cloud Computing", IEEE Systems Journal, vol. 13, no. 3, pp. 2739-2750, Sept. 2019.

[15] J. Sun, L. Ren, S. Wang and X. Yao, "Multi-Keyword Searchable and Data Verifiable Attribute-Based Encryption Scheme for Cloud Storage", in IEEE Access, vol. 7, pp. 66655-66667, 2019.

[16] R. Ahuja and S. K. Mohanty, "A Scalable Attribute-Based Access Control Scheme with Flexible Delegation cum Sharing of Access Privileges for Cloud Storage", IEEE Transactions on Cloud Computing, vol. 8, no. 1, pp. 32-44, 1 Jan.-March 2020.

[17] M. Ali, M. -R. Sadeghi and X. Liu, "Lightweight Revocable Hierarchical Attribute-Based Encryption for Internet of Things", IEEE Access, vol. 8, pp. 23951-23964, 2020.

[18] M. Morales-Sandoval, M. H. Cabello, H. M. Marin-Castro and J. L. G. Compean, "Attribute-Based Encryption Approach for Storage, Sharing and Retrieval of Encrypted Data in the Cloud", IEEE Access, vol. 8, pp. 170101-170116, 2020.

[19] D. Tiwari, A. Singh, A. Prabhakar, "Performance Analysis of AES, RSA and Hashing Algorithm Using Web Technology", Computing Algorithms with Applications in Engineering, Vol 18, pp 413, 2020.

[20] M. Rasori, P. Perazzo, G. Dini and S. Yu, "Indirect Revocable KP-ABE With Revocation Undoing Resistance", IEEE Transactions on Services

Computing, vol. 15, no. 5, pp. 2854-2868, 1 Sept.-Oct. 2022.

[21] A. Altigani, S. Hasan, B. Barry, S. Naserelden, M. A. Elsadig and H. T. Elshoush, "A Polymorphic Advanced Encryption Standard – A Novel Approach", IEEE Access, vol. 9, pp. 20191-20207, 2021.

[22] M. R. Khan et al., "Analysis of Elliptic Curve Cryptography & RSA", Journal of ICT Standardization, vol. 11, no. 4, pp. 355-378, November 2023.

[23] I. Mustafa et al., "A Lightweight Post-Quantum Lattice-Based RSA for Secure Communications", IEEE Access, vol. 8, pp. 99273-99285, 2020.

[24] B. Hong, J. Chen, K. Zhang and H. Qian, "Multi-Authority Non-Monotonic KP-ABE With Cryptographic Reverse Firewall", IEEE Access, vol. 7, pp. 159002-159012, 2019.

[25] Z. Li, W. Li, Z. Jin, H. Zhang and Q. Wen, "An Efficient ABE Scheme With Verifiable Outsourced Encryption and Decryption", IEEE Access, vol. 7, pp. 29023-29037, 2019.

[26] N. Chen, J. Li, Y. Zhang and Y. Guo, "Efficient CP-ABE Scheme With Shared Decryption in Cloud Storage", IEEE Transactions on Computers, vol. 71, no. 1, pp. 175-184, 1 Jan, 2022.

[27] S. Das and S. Namasudra, "Multiauthority CP-ABE-based Access Control Model for IoT-enabled Healthcare Infrastructure", IEEE Transactions on Industrial Informatics, vol. 19, no. 1, pp. 821-829, Jan, 2023

[28] Y. Zhang, J. Li and H. Yan, "Constant Size Ciphertext Distributed CP-ABE Scheme With Privacy Protection and Fully Hiding Access Structure", IEEE Access, vol. 7, pp. 47982-47990, 2019