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

HAP-BB: Hidden Access Policy-Based Cloud Data Sharing with Block Design and Blockchain Methodology

A. C. Ashmita, Dr. C. Yamini

Submitted: 23/05/2023 **Revised:** 13/07/2023 **Accepted:** 30/07/2023

Abstract*:* Policy on Ciphertext Attribute-based One of the best ways to manage who may access what data in the cloud is via encryption (CP-ABE). Key management becomes much easier when multiple users' attributes are taken into account while accessing shared data. The inability to safeguard policy data is a major issue with CP-ABE. In addition, data owners would incur prohibitive communication and computation costs due to the need to re-encrypt and transfer fresh data back to the cloud if data is often modified. A framework for accessing and exchanging data in the cloud, the HAP-BB, was suggested in this paper. Three approaches have been used in the HAP-BB architecture. A first method is HAP-ABE (Hidden Access-Policy Attribute-Based Encryption), which encrypts plaintext and communicates it with the ciphertext using an access policy. Critical information concealed from unreliable parties or system users in a real-world system's policies. The HAP-ABE access control system has a severe data limitation due to hidden features with Boolean architecture. The second one is blocked design-based key agreement protocol.Finally, we focused on blockchain-with intrusion detection and firewall security through a study on cloud storage administration models and blockchain innovation. The experimental results are compared with various existing methods.

Keywords: *Block design, Block Chain, Cloud data sharing, CP-ABE, HAP-ABE*

1. Introduction

Using cloud computing, a Cloud Service Provider (CSP) may efficiently handle and store data on behalf of a worldwide customer base. Cloud data storage is a cinch since clients don't have to worry about setting up and maintaining physical infrastructure [1]. Cloud computing is great but has new security concerns for user data since it is so powerful. Users lose ownership of their outsourced data when they no longer have physical access to it. As a consequence, the privacy and security of users' data are in danger [2].

As a result, data owners who outsource their data to the cloud need high levels of security and anonymity [3]. On the other hand, traditional cryptographic primitives cannot protect data directly. For some time now, researchers have been working feverishly on privacy and security issues related to sharing remotely stored data under various security models and systems (e.g., 2, 3, and 4). These studies primarily focused on protecting the user's privacy while achieving the necessary security level without burdening the user at the decrypted stage with excessive complexity. The problem can be solved using KP-ABE for access control and HAP-ABE for data security based on key policy attribute-based encryption. KP-ABE methods [4, 5] disclose particular features of cloud user access, indicating that they cannot entirely protect users' privacy and are not totally collusion resistant. [6], [9]

HAP-ABE-based systems are both inefficient and cumbersome. As a result, the problem of providing both private and effective cloud data-sharing services is still open [10]. The following conditions must be met to build a successful cloud computing data-sharing service while protecting user privacy. Before allowing a user to access his cloud data, the data owner must determine whether or not to allow it. Second, the cloud should not have access to the personal information of its users. Finally, linked terminals with little processing capacity, such as smartphones and tablets, allow users to access shared data [13] [14]. Until now, these crucial elements of cloud sharing have eluded us. This article tackles these concerns and offers a HAP-ABE data exchange method that is both efficient and privacy-preserving. The attribute-based encryption method we use with MD5 and RSA algorithms protects the privacy and guarantees data confidentiality against the cloud ABE.

In contrast to KP-ABE-based methods, HAP-ABE incorporates a user Partial Private Key (PPK) tightly connected to the secret user key, allowing for completely secure collusion while still protecting user privacy [16]. Instead of increasing the number of user keys to minimizing key management problems, HAP-ABE does not utilize HAP-ABE. The key management protocol is verified with block design methodology. The cloud

Research Scholar,
1Department of Computer Science, SriRamakrishna College of Arts & Science for Women, Coimbatore,Tamilnadu, India.
2Associate Professor, Department of Computer Science, Sri Ramakrishna College of Arts & Science for Women, Coimbatore, Tamilnadu, India.

users' access control is often implemented in three stages: evidence, validation, approval, and blockchain architecture. Besides permitting only legitimate customers to enter, control ensures accountability: the ability to track which client did which action inside a framework [17]. In conventional access control frameworks, security administrators determine which users have access to a certain piece of data. Thus, these current frameworks are becoming more vulnerable to hacking and inexplicable disruptions [18-20].

The main contributions of this paper are as follows

- Cloud data has been shared with HAP-ABE
- Key management protocol has been verified with the Block Design method
- Data is secured with blockchain for security reasons.

The following paper is organized: Section II discusses several study strategies for detecting cloud access control and blockchain methods. Section III discusses the proposed method in detail. Section IV illustrates the experimental setup and the discussion, and Section V has the conclusion and future efforts.

2. Background Study

Anjali, R. S., & Ravikumar, A. [1] Cloud infrastructure enables authenticated cloud customers to access many services that are transferred and accumulated in the cloud. An auditing mechanism conducted with the assistance of a third-party inspector to protect data integrity and prevent unauthorized users from accessing the users' sensitive data. The auditing mechanism assists consumers in determining if outsourced services are stable in the cloud. During auditing, the public verifier will examine the hidden data material and user identity protection.

Chandanapriya, E., & Murali, G. [2] There are no pairing procedures in the very efficient system. Verification of forward protection and complete anonymity is possible. The hidden key is a single integer, and the only operation required to update the key is to multiply it by itself. Improved dependability, massive capacity, and data exchange are all results of using the technology in a multi-cloud system. Many other real-world applications might benefit greatly from the author's approach. This is particularly true for ad-hoc networks, e-commerce, and smart grids, all of which need user authentication and privacy. The foundation of our present plan is proving protection.

Dhivya V. et al. [3] multi-owner cloud is becoming more popular for stable data sharing within a community. Customers in a multi-owner cloud may encrypt their data using their private key method and then distribute it in an unsecured cloud. The signature for cloud authentication (in a group) is created using these private keys; these are called ring signatures.
In (J. Lai et al. [6]), proposed a secure ciphertext method to keep CP-ABE a secret. Inner-product predicate encryption (PE) is their method, and Composite order bilinear groups are used. In the PE system, each ciphertext is given a unique anonymity predicate stored in the user keys. On the other hand, complexity results from both the number of predicates and the Composite order groups.

Zhao et al. [7] use the CPABE and Attribute-Based Signatures (ABS) to provide a safe way to share data. They've found a way to control both read and write access using their approach. ABS' verification characteristics specify the write permission attributes in the Tsign format, which is signature-based. Users who want to make changes to a file and then re-upload it to the cloud must first get permission from the cloud server.

Fan et al. [18] proposed a dynamic membership management arbitrary-state ABE to solve the issue. This article allows users to dynamically join, quit, and modify their characteristics and provides a high degree of attribute limitation flexibility. A person can join and leave an ABE system and change their characteristics and values along with those traits. No one else needs to update their private key when a member joins, quits, or changes their profile.

a) Problem definition

Clients keep massive amounts of sensitive data in the cloud. In our existing framework, sharing sensitive data will aid businesses in lowering the cost of providing personalized services to customers and provide certain advantages, including data services. In any event, secure data exchange is fraught with danger. As a result, security is not updated in its current state. When multilevel execution is ensured, a vulnerability occurs. For example (the transferred infections and Trojans, much after the client may constantly attempt to transfer the infection document). This condition ensures that a server is kept busy. By implementing a nasty client-blocking concept, we are resolving this problem. Because the cloud is a multi-client open-source, unauthorized access might misuse data. View permission is allowed under our proposed framework; for example, records requesting approval are delivered by email to the authorized person's email address, allowing the approved individual to access the documents.

3. Materials and Methods

We proposed HAP-BB Framework for securing the cloud data and access control policy.This framework has three main parts: hidden access policy, block design, and blockchain methodology.

3.1 HAP-ABE

The proposed HAP-ABE scheme consists of four algorithms: System Setup, KeyGen, Encryption, and Decryption.

3.1.1 Setup (l) ⟶ (HK, PPK)

The public HAP-ABE for Cloud Storage key is used for data encryption, while the system master key is utilised to produce private keys for registered users. The Setup method accepts a set of system security settings as input and returns the system Hash Key HK and system Partial Private Key PPK.

3.1.2 KeyGen (A, HK) → PPK using (MD5) Hash Code

In order to provide consumers with Hash Code, the Key Gen algorithm is executed. Outputs the user's precise private key

based on their attribute set A, using the system partial private key PPK as input.

3.1.3 Encryption (PPK, HK) → CT (RSA) (Re-Encryption)

sA message M and the system public key PK is encrypted using the Re-Encryption algorithm according to a Hidden Access Structure (HAS) that are provided. Only a user's set of attributes that are compatible with the HAS may decode the ciphertext CT that it generates. A false structure must be used to accomplish the goal of access privacy preservation. The ciphertext CT is appended after its construction. Then, the cloud would be used for storing and distributing both HAS and CT.

3.1.4 Decryption (CT, HK, PPK)

Using the user's secret/private key SK, the decryption process converts the ciphertext CT to plaintext M. In order to decipher and retrieve the original message M, the user's attribute set A_j has to meet the anonymous access policy AS, which isA_j,A_S that is associated to the ciphertext CT.

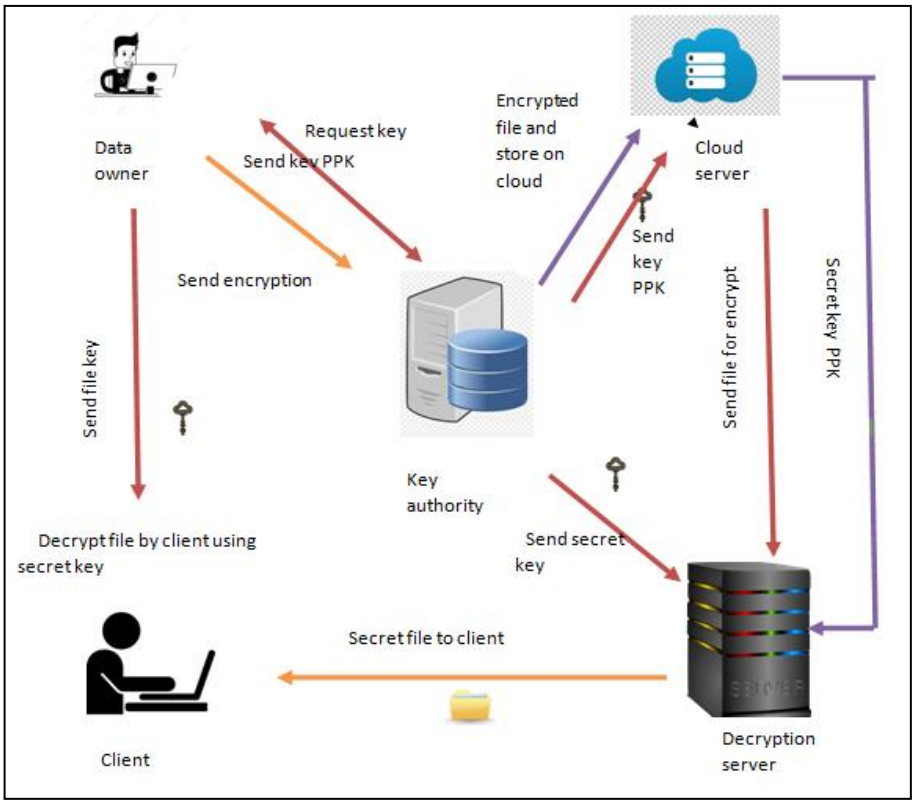

Figure 1: HAP-ABE Access Architecture

3.2 Block Design-Based Key Agreement Protocol

3.2.1 Initial Phase

An organization's Third Party Auditor (TPA) role is to establish the protocol's settings and disseminate the protocol's private key to users. The TPA discloses the following during the key generation portion of the protocol while protecting his private key2 $Z\ q$: Weil pairing parameters G,G_1,G_2, and e are determined in the first description, and p and q are positive integers referred by Zhou, T. et al. (2023).

InH1 and H2 of the supplementary hash functions convert arbitrary lengths to non-zero values in G1 and an integer, respectively. Using H1(ID_i) and $S_i = sH1$, our block-based key agreement approach notates the mapping between the public and private keys of the participants (ID_i). Each person in the research has a special ID, represented here as ID_i 2 $f0;\ 1g$. Public key ei and private key di, where ei is the public key and di is the private key, are chosen by the TPA for each participant in the RSA authentication process. The resultant pair (i.e., n), where n is the product of two big prime numbers, is then distributed to all participants. $Y_i = H2(ID_i)$, $X_i = (Y_i)d_i$ and a hidden number are the best approximations I can make at the moment($d_i;\ X_i$).

3.2.2 Key Agreement Phase

There must be two rounds of key agreement before a large group of people may agree on a common conference key. Structure E of the $(v;\ k + 1;\ 1)$-design for the concept of group data sharing determines the format of messages sent and received referred by Yang, Z. et al. (2023).

In the first round, a secret key of random integer ri is chosen, and each participant generates a portion of the conference key by computing Mi = e(G; Eirini). Timestamp t_i , where $X_i = Y\ d_i\ I\ wi = H2(ID_i)$, and authentication key $Y_i = H2(ID_i)$ are used (Mi;ti). Participants will relay the information that$D_j = fY_j\ ;\ (M_j)ei;\ T_j;\ if_j >= E_i$. And because there are people in each Ei block, I 2 Ei follows. However, there is no need for the participant to get a message from himself. $D_j = fY_j\ ;\ (M_j)e_i\ ;\ T_j$; tjg is the message that participantj sends to

participants if_j 2 $E_i(j$ 6 = $I.A$ $(v; k + 1; 1)$ design has four possible mathematical representations that correspond to the four distinct phases of crucial agreement in Round 1.

Case 1:

One of the participants (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, and 10) must send messages to the other.

Case 2:

Communicating with participant i (i k) requires messages from participant$(j = mk + 1 + MODk$ $(i$ 1) $(m$ 1); j 6 = I.

Case 3:

Communication between participant0 and participant j $(j = b(i$ 1) = kc $k + m; j$ 6 = I is required for participant $(i = Em; m)$.

Case 4:

Out of the remaining k2 k individuals, each one must receive a message from participant$j$, where j is equal to mk + 1 + MOD_k $(mx$ x $m + r)$; j 6 = I and r is one of 2, 3, 4... k 1; k.

Participants decrypt communications by calculating Eq. 1 after receiving k messages from their intended message senders that were donated to produce a shared conference key.

$M_j = [(M_j)^{ei}]^{di}, j \in E_i - \{i\}$ ------- (1

Where d_i is a participant's private access code, participanti calculates Tejj =M w j j to confirm the user's identity upon each participation. $If\ Tej_j = M_j = Y_j$, where $w_j = H_2$, then participant j may authenticate participant j $(M_j; t_j)$. In addition, $Ci; j$ determined with the help of Eq. 2, and it will be utilized to generate a shared conference key for player j in the next Round 2.

$c_{i,j} = \prod_{x \in E_i - \{j\}} M_x$ -------- (2)

Round 2: If I 2 Ej, the other participant will get the message$E_j; I = fY_j ; (C_j; i)e_i ; (M_j)e_i; T_j$, where C_j I am the shared conference key generated by the participants. If k people use C_j as their conference key, I'll send k messages to each of them. Similar to the first round, the user verifies the authentication equation$Tej_j = M_j = Y_j$. Assuming the equation holds, participanti may confirm the identity of participants, but not vice versa. As a result, everyone's share of the conference key is determined by a formula of Eq. 5.

$k = m_i(\prod_{j \text{ such that } i \in E_j} C_{j,i})$ --------- (3)

$= e(g, e_i r_i s_i).(\prod_{j \text{ such that } i \in E_j} C_{j,i})$ -------- (4)

$= e(g, \sum_{i=0}^{u-1} e_i r_i s_i)$ -------- (5)

Multiple people in the same group may generate a common meeting key using the block design-based key agreement method.

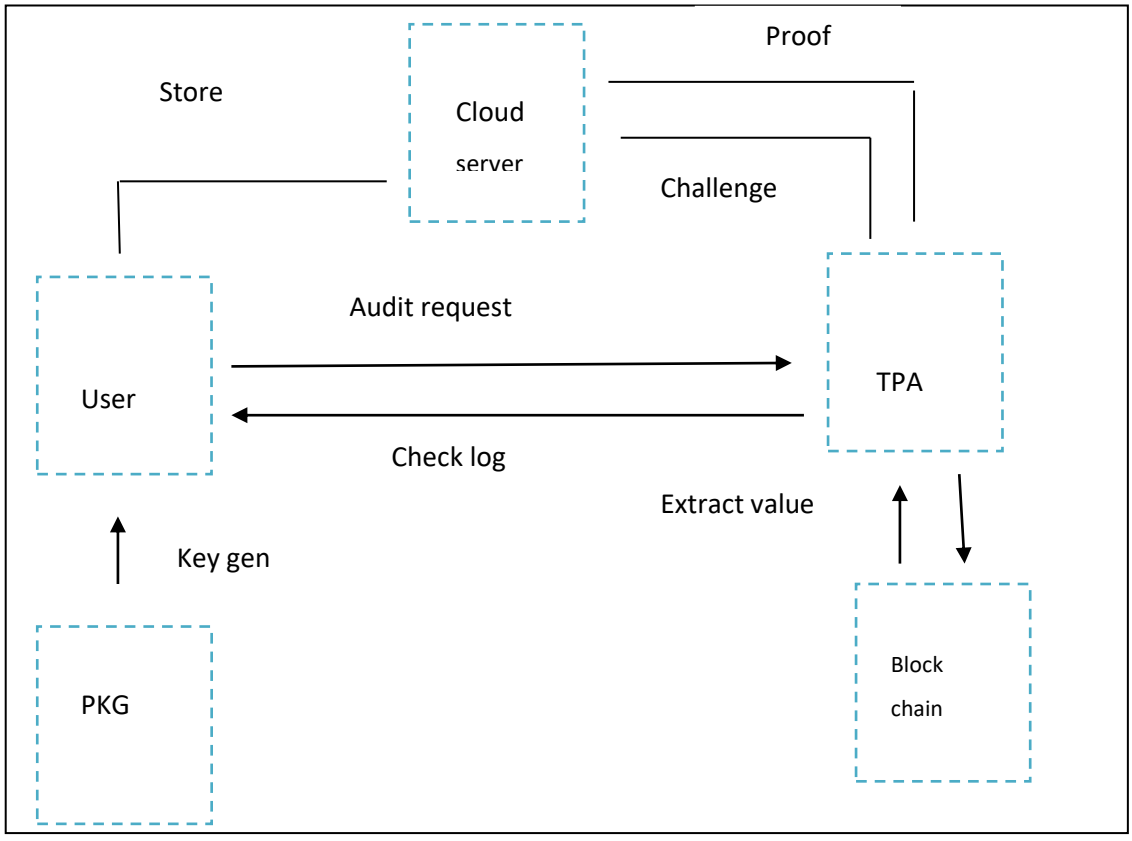

Figure 2: Block design and block chain based data sharing and auditing

3.3 Proposed Construction of BCPA

The BCPA's comprehensive design and how it works with the standard methods used in public auditing today.

Figure 2 shows how we swap a traditional TPA with a blockchain network. There are two main stages to most procedures: the setup and audit phases. Operations performed by various entities in a BCPA are similar to existing schemes, except for the Challen, Audit, and CheckLog algorithms.

Much like other public auditing programmes, this one follows a similar format. At this point, the PKG starts the system and generates secret keys for each user by activating the Setup and KeyGen algorithms, respectively. The TagGen method makes it possible to outsource the task of assigning data blocks and tags to the CS. Initialization (Specific; Parameters; Physiological; Molecular) In order to construct the bilinear map e: D1 D1! D2, the PKG uses P as the generator of G1 and produces two cyclic groups of order p. Following this, we choose MSK = s 2 Zp and derive PK = Q = sP, where P is the public key. Also, it details a set of hash functions, including H_1; H_2: f_0; $1g_! D_1$, h: D1! Zp, and$H$: f_0; $1g_! Z_p$. A state parameter, w, is also generated. There is a public parameter, and its value is:

$Params = (D_1; D_2; e; p; P; C_1; C_2; C; h)$: ------- (6)

This is a KeyGen $(P_K; MSK; I_D)$! Identifier: (SKU; PKu). To do this, it takes the user's ID and hashes it into two separate components, Pu;0 = C1(ID; 0) and Pu;1 = C1(ID; 1), and then calculates Qu;0 = sPu;0 and Qu;1 = sPu;1. SKu = fQu;0; Qu;1g is the private key, while PKu = fID; Pu;0; Pu;1g is the public. The TagGen(Params; F; SKu)! Setting. Data file F is divided by the user into n chunks, F = m1||m2||mn||.

Settag = $f\left(S_j ; T_j\right)g_j 2[1; n]$ is the set of all authentication tags. Next, the user deletes the F file from local storage and transmits fF, Settagg, to the CS.

Approaches to Validation A blockchain examination has been requested by a user. The CS uses the ProofGen algorithm to create the proof data whenever it gets a challenge message from the blockchain.

IV Results and Discussion

The proposed method has implemented by using python programming language with django framework.

Approval is the process that allows access to a verified user after authentication, which verifies the user's identity. Typical entry points for attackers include security holes in the system's verification and approval processes. It is possible for the third party to harm the system by evading its permission and verification processes.

Table 1: Comparison of computation complexity of various schemes

Operation	**Time-complexity**	**HAP-BB**
File Upload	*O(n)*	*O(n)*
Efficient File Access	*O(n)*	*O(n)*
Efficient Auditing	-	*O(n)*
File Deletion	-	*O(1)*

Comparing ElGamal with the suggested HAP-BB system, Table 2 displays the average calculation time for different stages. Table 2 shows that compared to the ElGamal scheme, the HAP-BB scheme has a shorter calculation time, making it the superior choice.

Table 2: Average computation time for various phases in ElGamal scheme and HAP-BB scheme

	File Size (MB)	**ElGamal (Sec)**	**HAP-BB(Sec)**
File upload Total time (Encryption+	1	0.05	0.07
	5	0.36	0.041
	10	0.46	0.152
	20	0.37	0. 131
	50	0.62	0.138
File Download Total time (Decryption+	1	0.15	0.06
	5	0.66	0.45
	10	0.71	0.34
	20	0.73	0.45
	50	0.66	0.56

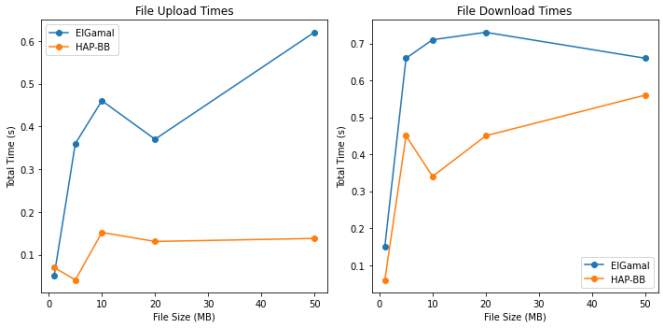

Figure 3: Average computation time for various phases

The table 2 figure 3 data represents the total time required for file upload and download operations, including both encryption and data transmission for upload and decryption and data transmission for download, measured in seconds. For file upload, the ElGamal encryption method exhibits varying times, increasing from 0.05 seconds for a 1MB file to 0.62 seconds for a 50MB file, while HAP-BB shows a different trend, starting at 0.07 seconds for 1MB and decreasing to 0.138 seconds for a 50MB file. In file download scenarios, ElGamal consistently demonstrates higher times, ranging from 0.15 seconds for a 1MB file to 0.66 seconds for a 50MB file, whereas HAP-BB follows a fluctuating pattern, starting at 0.06 seconds and reaching 0.56 seconds for the same file sizes. These results highlight the performance differences between ElGamal and HAP-BB encryption methods across various file sizes in terms of total time for both upload and download operations.

Table 3: Average Computation Time of various phases of File Upload

FileSize (KB)	Total Time (Sec)	Data Transmission Time(Sec)	Encryption + KG Time (Sec)	key management (Sec)
1	00.048210	00.034410	00.01130	00.00250
10	00.064920	00.043120	00.01720	00.00460
50	00.09691	00.068210	00.02220	00.00650
100	00.13381	00.071110	00.05530	00.00740
1000	00.17121	00.084510	00.07740	00.00930

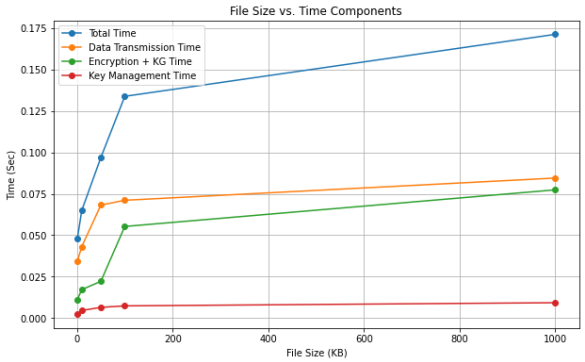

Figure 4: Average Computation Time of various phases of File Upload

The table 3 and figure 4 shows data outlines the performance metrics for file operations at different file sizes, measured in kilobytes. Starting at 0.04821 seconds for a 1KB file and increasing to 0.17121 seconds for a 1000KB (1MB) file, the total time for file processing grows steadily with greater file sizes. Time spent transmitting data, encrypting it, creating keys, and managing those keys all add up to this total. Notably, data transmission time is a significant factor, escalating from 0.03441 seconds for 1KB to 0.08451 seconds for 1000KB. Encryption and key generation time also contribute to the overall duration, with values increasing as file sizes grow. Key management time, however, remains relatively small across all file sizes, indicating that the primary contributors to the total time are data transmission, encryption, and key generation. According to the statistics, the file processing system's performance is affected by file size, and the particular time components involved in the operation are illuminated.

Table 4: Average Computation of various phases of file Download

File Size (KB)	Total Time (Sec)	Data transmission Time(Sec)	Decryption + KG Time (Sec)	Key Management (Sec)
1	00.0608	00.03820	00.02150	00.00110
10	00.0879	00.04610	00.03760	00.00420
50	00.16581	00.058210	00.05250	00.05510
100	00.19393	00.06130	00.06540	00.067230
1000	00.26101	00.074410	00.09730	00.08930

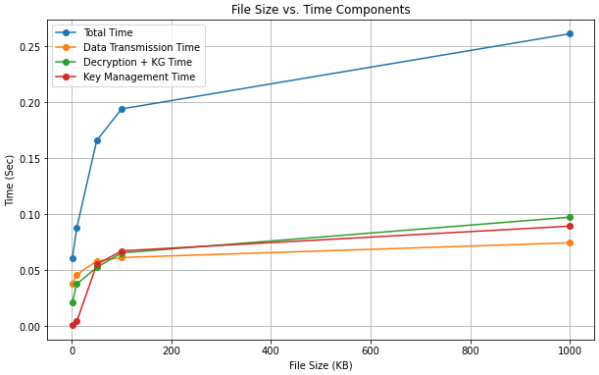

Figure 5: Average Computation of various phases of file Download

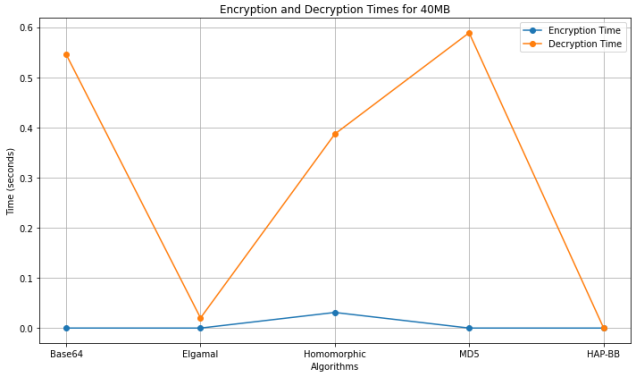

Figure 6: Encryption and Decryption time comparison chart

The table 4 and figure 5 outlines the performance metrics for file operations at different file sizes, measured in kilobytes. The total time for file processing increases gradually with larger file sizes, ranging from 0.04821 seconds for a 1KB file to 0.17121 seconds for a 1000KB (1MB) file. This total time comprises data transmission time, encryption, key generation time, and key management time. Notably, data transmission time is a significant factor, escalating from 0.03441 seconds for 1KB to 0.08451 seconds for 1000KB. Encryption and key generation time also contribute to the overall duration, with values increasing as file sizes grow. Key management time, however, remains relatively small across all file sizes, indicating that the primary contributors to the total time are data transmission, encryption, and key generation. The data underscores the impact of file size on the efficiency of the file processing system, shedding light on the specific time components involved in the operation.

Table 5 proposes three methods that, when combined, may improve performance. An audit table is maintained to store the data needed to conduct an audit of private information. The amount of squares placed in the cloud properly corresponds to the audit table's capacity overhead. In an effort to reduce the framework's efficiency, an audit is being prepared to complete various probabilities of damage detection. Regardless, the intended confirmation is an effort to differentiate early damage detection.

Table 5: Comparison for various algorithms

Algorithm	File Size	Encryption Time	Decryption Time
Base64	40MB	00.000090	00.54541420
Elgamal	40MB	00.000040	00.02020030
Homomorphic	40MB	00.03120	00.38695880
MD5	40MB	00.000080	00.58875750
HAP-BB	40MB	00.000030	00.0010400

The results of using the hybrid algorithm for encryption and decryption in Table 5 the methods included Base64, Elgamal, homomorphic, and MD5.

Threats to Integrity: The confidentiality of public documents faces two main threats. As a first step, a malicious actor might try to get unauthorised access to shared data. Second, the cloud provider could inadvertently damage or erase data stored in its systems as a result of hardware issues or human mistake. Even worse, the cloud provider is profit-driven, so it may not be able to alert users to the possibility of data modification in order to safeguard the data's authenticity and avoid losing revenue.

Threats of Privacy: The signer's name on each shared data block is private and confidential to the party. During the auditing procedure, a public verifier, who is only permitted to verify the correctness of shared data integrity, can attempt to disclose the signer's identity on each block of shared data using verification metadata. Suppose the public verifier has shown the identity of the signer on each block. In that case, it can clearly differentiate between a highvaluetarget (a specific user in the community or a special block of shared data) and others.

V. Conclusion

We proposed HAP-BB framework for cloud data sharing with efficient access methodology. This type of data sharing uses the features of the data-sharing system to provide users fine-grained control over the data they have access to. Key escrow problems are eliminated since the private user keys are generated using a secure PPK. The proposed method has three stage, First one is HAP-ABE (Hidden Access-Policy Attribute-Based Encryption) based access control; access policy is often used to encrypt plain data and is communicated with the ciphertext. Policies in a real-world system may include critical information that is hidden from untrustworthy parties or even system users. The HAP-ABE access control system has a severe data limitation due to hidden features with Boolean architecture. Second one is block design based key agreement protocol. Finally, we focused on blockchain-with intrusion detection and firewall security through a study on cloud storage administration models and blockchain innovation.A comparison between the new system and the existing one is shown in experimental graphs. As a result, the proposed approach takes less time to encrypt and saves cloud service provider storage space. The findings show the suggested work's great efficiency and security.

VI Reference

[1] Anjali, R. S., & Ravikumar, A. (2016). Preserving privacy in public auditing for shared cloud data. 2016 International Conference on Inventive Computation Technologies (ICICT). doi:10.1109/inventive.2016.7824838

[2] Chandanapriya, E., & Murali, G. (2016). Effective data sharing using advanced ring signature with forward security. 2016 International Conference on Communication and Electronics Systems (ICCES). doi:10.1109/cesys.2016.7889937

[3] Dhivya V., Anandakumar H, & Sivakumar M. (2015). An effective group formation in the cloud based on Ring signature. 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO). doi:10.1109/isco.2015.7282366

[4] Nishide, T., Yoneyama, K., and Ohta, K., 2008. Attributebased encryption with partially hidden encryptorspecified access structures. In Proceedings of Applied Cryptography and Network Security, ACNS'08. LCNS, Vol.5037, pages 111-129. Springer.

[5] Katz, J., Sahai, A., and Waters, B., 2008. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In Proceedings of International Conference on the Theory and Applications of Cryptographic Techniques, Eurocrypt 2008. LNCS, Vol 4965. pages 146-162, Springer.

[6] Lai, J., Deng, R. H., and Li, Y., 2011. Fully Secure Cipertext-Policy Hiding CP-ABE, In Proceedings of the 7th International Conference on Information Security Practice and Experience, ISPEC'11, pages 24- 39, Springer.

[7] Zhao, F., Nishide, T., and Sakurai, K., 2011. Realizing Fine-Grained and Flexible Access Control to Outsourced Data with Attribute-Based Cryptosystems, In Proceedings of 7th International Conference of Information Security Practice and Experience, ISPEC'11, papges 83-97, Springer.

[8] Ruj, S., Stojmenovic, M., and Nayak, A., 2012. Privacy Preserving Access Control with Authentication for Securing Data in Clouds, In Proceedings of 12th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, CCGrid 2012, pages 556-563, IEEE.

[9] Li, M., Yu, S., Zheng, Y., Ren, K, and Lou, W., 2012. Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption, In IEEE Transactions on Parallel and Distributed Systems. Volume 24, Issue: 1, pages 131-143, IEEE

[10] Fugkeaw, S. and H. Sato, 2015. An extended CP-ABE based Access control model for data outsourced in the cloud, In Proceedings of IEEE International Workshop on Middleware for Cyber Security, Cloud Computing and Internetworking, MidCCI 2015, IEEE.

[11] B. Waters and A. Sahai "Fuzzy identity-based encryption" Proc. 24th Annu. Int. Conf. Theory Appl. Cryptograph. Techn., pp. 457-473, 2005

[12] O. Pandey, V. Goyal, B. Waters, and A. Sahai. "Attribute-Based Encryption for Fine-grained Access Control of Encrypted Data," Proc. 13th ACM Conf. Computer and Comm. Security (CCS' 06), pp. 89-98, 2006

[13] A. Sahai, B. Waters and J. Bethencourt "Ciphertext-policy attribute-based encryption" Proc. IEEE Symp. Secure. Privacy, pp. 321-334, May 2007

[14] S. S. M. Chow and M. Chase "Improving privacy and security in multi-authority attribute-based encryption" Proc. 16th ACM Conf. Comput. Commun. Secur., pp. 121-130, 2009.

[15] J. Hur "Improving security and efficiency in attribute-based data sharing" IEEE Trans. Knowl. Data Eng., vol. 25, no. 10, pp. 2271-2282, Oct. 2013

[16] X. Chen, X. Xie, H. Ma and J. Li "An efficient ciphertext-policy attribute-based access control towards revocation in cloud computing" J. Universal Comput. Sci., vol. 19, no. 16, pp. 2349-2367, Oct. 2013

[17] S. Katzenbeisser, S. Müller, and C. Eckert, "Distributed attribute-based encryption," in Proc. 11th Int.Conf. Inf. Secure Cryptol, pp.20– 36, 2009.

[18] C.-I. Fan, H.-M. Ruan and V. S.-M. Huang "Arbitrary-state attribute-based encryption with dynamic membership" IEEE Trans. Comput., vol. 63, no. 8, pp. 1951-1961, Aug. 2014

[19] Yang, Z., Wang, Z., Qiu, F., & Li, F. (2023). A group key agreement protocol based on ecdh and short signature. Journal of Information Security and Applications, 72, 103388.

[20] Zhou, T., Wang, C., Zheng, W., & Tan, H. (2023). Secure and efficient authenticated group key agreement protocol for AI-based automation systems. ISA transactions.