Vol. 06, No. 3 (2024) 1327-1336, doi: 10.24874/PES.SI.24.03.015

Proceedings on Engineering Sciences

www.pesjournal.net

IDENTITY-BASED PRIVACY-PRESERVING ANONYMOUS AUTHENTICATION ACCESS CONTROL FOR SECURE CLOUD COMPUTING

N. Praneetha
S. Srinivasa Rao[1]
Maheswara Rao V V R
I. S. Siva Rao
P. S. Brahmanandam

Received 28.10.2023.
Received in revised form 15.11.2023.
Accepted 18.12.2023.
UDC – 004.451.52.056.5

Keywords:

Privacy-preserving cloud computing, Secure Data sharing, Identity-based Hierarchal Cryptographic, Access Control Policy, Secure Authentication.

ABSTRACT

The storage of data with access controllability in sharing data between multiple users in a dispersed environment is now the most pressing issue in cloud computing. Cloud computing's unique feature allows its subscribers to exchange and manage their data with one another safely. However, privacy is essential for all cloud users to communicate freely and openly, as data is accessed by unwanted parties in the cloud. Cloud services have employed numerous security-related techniques for efficient and safe user data exchange. These methods provide efficient, flexible, and reliable access control rules between users when exchanging data. But, they each have advantages and disadvantages concerning key generation and data security. To address the need for adaptable, scalable, and trustworthy access control while exchanging data across a dispersed network, this research proposes a novel access control-based privacy-preserving approach. In terms of cipher text and critical policy security, this method is an extension of attribute-based encryption and the only difference is that with cloud computing, users hierarchically share data, and access control policy amongst shared users is evaluated efficiently. Our method includes sophisticated security features like the revocation of users and access rights of users for outsourced data in the cloud. It also provides scalability and dependability in producing critical structures with dynamic qualities. Experimental results demonstrate that the suggested method improves the current system in terms of cloud data sharing efficiency, scalability, and dependability.

1. INTRODUCTION

Distributed computing is an innovative, constantly evolving science-related application that consists of interconnected data sets with shared adaptability, skill in sharing, and web-dependent client demands. In addition, it is a promising approach to determine capacity limits and should be used everywhere. It uses robust, flexible resources in the cloud to lessen the computational expenditure incurred by data owners while sharing data via cloud clients. Distributed computing relies on cloud service providers (CSPs), who offer various cloud-based services, such as Software As A Service (SAAS),

[1] Corresponding author: S. Srinivasa Rao
srinu1479cse@kluniversity.in

Platform As A Service (PAAS), and Infrastructure As A Service (IAAS), to their users.

Customers may use these services to monitor the effectiveness of cloud-based project management and collaboration tools. Due to the ever-changing user base and unique security concerns of the cloud, effective and valuable cloud services for users must be evaluated. CSP stores and conducts sophisticated operations to put away information for a subset of businesses that deal with cloud employees. Critical resources are present in the information that CSP processes, and as a result, CSP can cause unexpectedly large amounts of data loss. Therefore, while considering cloud security, it is essential to consider concerns about keeping private data safe.

While data privacy isn't strictly necessary, reasonably achieved access control is a highly desired feature in assistance-oriented distributed computing. Some medical services and related organizations evaluate specific components to address a variety of tasks to a group of clients rather than creating multiple copies for each group member (Ahuja and Mohanty, 2020; Rasori et al, 2022). It's often more efficient to use shared admittance privileges (SAP) to give everyone in the group access to the information they need, with or without the individual's permission. First, SAP uses and isolates information benefits for all of the group's users, and second, it avoids wasting computing resources by confirming information that isn't necessary. Standard practices for data sharing in the cloud include examining the merits of several approaches, such as assigning clients individually or in groups.

Attribute-based encryption (ABE) (Wang et al, 2011) describes secure authentication. However, this methodology employs a symmetric key-based cryptographic approach, which does not provide efficient authentication concerning key encryption and, therefore, the traditional workhorse behind access control policies in the cloud. If we expand the number of users in the cloud, the strategy taken by Zhao et al. (2017), which employs a distribution of keys, will only enable single-key communication. As a result, it will not be able to support multiple key generations in a distributed setting.

While the solution to access control is decentralized, it does not support authenticated users, it does not allow users to read and write files in the cloud, and it does not restrict access to files based on who created them (Ruj et al., 2004). Using the benefits and drawbacks of the methods mentioned, we may improve distributed computing by expanding the privacy-related features and enabling the authentication-related features related to the access control policy for data sharing among all users.

To facilitate adaptable, scalable, and trustworthy access control during data sharing in a distributed setting, this research suggests a novel access control privacy-preserving approach. The proposed method is resilient to various relay attacks, meaning the user can replace the old file with one that includes both reading and writing. This method can also be used to revoke user access, i.e., deny access to previously authorized users before granting them access with modified permissions.

The key aims of our proposed strategy are as follows:

a) When validating user parameters in the cloud
    i) Only authorized users should have access to the corresponding access control data
    ii) We verify and change the identities of any users with access to the data, whether authorized or not.
b) We keep the design split for efficient key management so that no two users may access the data simultaneously or "co-exist" and avoid collisions.
c) The cloud allows for numerous read/write state operations; if a user's access is revoked, that person is prevented from accessing any data.
d) Experiments demonstrate that the suggested technique is scalable and dependable for exchanging data in the cloud, using a variety of performance indicators not previously considered.

2. RECENT RESEARCH STUDIES-LITERATURE SURVEY

L The researcher assessed attribute-based encryption (ABE) and gave a concise outline of the ASBE. From that point forward, the current access control methods were investigated, dependent on ABE. The possibility of ABE was first proposed by Sahai and Waters (2005) as another method for hazy character-based security. The essential issue with the arrangement is that its restricted semantics need to be impressible. A few drives have been continued in writing to settle the impressibility issue. In the ABE conspire, cipher texts are not getting to a specific client as in the customary local area key cryptography. Both cipher texts and clients' decoding significant variables may be related to the characteristics or an arrangement of highlights.

A client can unscramble a cipher text if there is coordination between the decrypted key and encrypted text. ABE strategies are arranged into key-approach trait-based security (KP-ABE) and cipher text-strategy characteristic-based security (CP-ABE) in light of how ascribes and plans are related to encrypted text and clients' unscrambling keys. In a KP-ABE plan (Li et al., 2017) a cipher text is related to many highlights, and a client's unscrambling key is associated with a single topic bush availability system. If the highlights on the cipher text satisfy the bush openness structure, then the client can unscramble the cipher text.

In a Ciphertext-Policy Attribute-Based Encryption (CP-ABE), the parts of encrypted texts and unscrambling significant elements are exchanged (Helil & Rahman, 2017); the cipher text is obtained with a bush availability plan chosen by an encrypt or, while the comparing decoding key is planned regarding a bunch of highlights. Given that the arrangement of highlights related to an unscrambling key meets the bush openness strategy for a given cipher text, the key can decode the cipher text. Since the significant client unscrambling factors are related to many highlights, CP-ABE is adroitly closer to customary access control plans like Role-Based Access Control (RBAC) (Alturi & Ferraiolo, 2011; Roslin Dayana & Shobha Rani, 2023).

In this way, CP-ABE is more feasible to implement for performing availability command over obtained data than KP-ABE. Nonetheless, essential CP-ABE strategies combined with secure operations to back up openness control in contemporary business environmental factors (Zhao and Wan, 2017), which need colossal adaptability and proficiency in indicating rules and dealing with client highlights (Samanthula et al., 2015). In a CP-ABE plan, the significant decoding factors help client credits be organized reasonably as just one set. Clients can utilize all potential highlights in just one set given in their privileged insights to satisfy rules. To fix this issue, Samanthula et al. (2015) implemented encrypted text trait set-based Encryption (CP-ASBE or ASBE for short). ASBE is an all-inclusive CP-ABE that orchestrates client highlights concerning network operations.

ASBE can execute incredible limitations on blending ascribes to satisfy an arrangement, which gives brilliant adaptability in openness control. In the recursive list of capabilities allotted to a client, highlights from a similar set can be blended rapidly. In contrast, highlights from better places must be joined by changing over items whose activity will be clarified later. Similar security operations create the highlights for students by them. Every understudy has a variety of attributes according to the courses they have completed. The specialist needs to have an arrangement. "Understudies who took a class that fulfils and upholds such an arrangement with CP-ABE are trying since an understudy might have taken a few projects and gained various evaluations.

The encryption should ensure the understudy cannot pick union highlights from better places to go around the arrangement. As suggested by (Samanthula et al., 2015), a few potential options with plain CP-ABE are portrayed, yet they need to be more adequate. In any case, by utilizing ASBE, the issue can be fixed by allocating a few standards to the arrangement of highlights in various sets. For each course, students get different standards for the highlights. Along these lines, ASBE can execute successful cipher text plan encryption for conditions where the current ABE strategies are insufficient. Besides, ASBE's capacity to give a few qualities to a similar component permits it to adequately determine the customer denial issue, which is a difficult recommendation in CP-ABE. The denial issue can be fixed rapidly by giving different diverse experiments.

Huang et al. (2017) recommended hierarchal ABE (HABE) to acquire the availability of fine-grained control in distributed storage space arrangements by blending hierarchical identity-based Encryption (HIBE) and CP-ABE. This method also maintains transparency while granting the thinking suppliers granular flexibility overestimations. However, as managed by a comparable area master, HABE employs a disjunctive ordinary structural plan and covers all the highlights in a single conjunctive statement. Therefore, certain area aces may administer a similar quality as suggested by particular rules that are attempting to be followed. Furthermore, according to ASBE, this plan does not support certain deserving duties and is unable to sufficiently assist with drug inclusion. In distributed computing, a new and more effective method is needed to provide a dynamic access control structure amongst users.

3. BASIC PRELIMINARIES

This section describes the essential preliminaries, assumptions, data owner, cloud server, and data sharer. The server relates to the cloud, explores data storage with different services, and provides efficient access control policies in the user's stored data concerning credentials on authorization. The server of the cloud acts as a semi-trusted device (Chen et al., 2019). It provides efficient access control on encrypted data with authorized secret key sharing on plain text stored in the cloud. The cloud server helps to manage all the authorization privileges with their registered user credentials related to revoked data files.

The data owner is registered as a cloud user who can store data in the cloud and share secret files with other users present in the cloud concerning authorized privileges and user-related credentials. The owner of data indexes and stores all the official credentials of users, issues related to credentials with authorized details to the sharer of data, and the owner of data revokes the user's data without any notification to the sharer of data.

Sharer of data can explore only encrypted data, which provides authorization to the entire files list, which can have efficient credentials, and also shows the tender proof by cloud server, which can have efficient credentials. By using encrypted files with their mask boundary code values, which transfer data with an authorized file key, the sharer of data should identify data related to sensitive data that contain delegated credentials. Basic notations used in the proposed implementation are described in Table 1.

Table 1. Shows the Basic notations used in this study

Symbol	Description
V_μ	Owner/user of vth user
X_j , X , L_j	jth attribute relations Key Distribution Center (KDC)
$l_j =\mid L_j \mid, I[j,u], I_u$	Number of claimed user attributes for encrypt/decrypt
$PK[j] / SK[j], sk_{i,v}$	Public key/secret key
H, H, MSG	Hash functions with message

Based on the above parameters, the following concepts are used in the proposed implementation, i.e., basic formats of access policies, access tree structure, and ABE. Let us discuss them in detail hereunder.

a) Basic format of access policies in cryptography

Basic access control policies are described in the following formats, which include attributes relate to Boolean functions, secret sharing linear schema and span programs relate to monotone.

An access tree structure converts Boolean functions.
For example, $((x_1 \wedge x_2 \wedge x_3) \vee (x_4 \wedge x_5) \wedge (x_5 \wedge x_6))$, $x_1, x_2, \ldots.$, x_6 are attributes.

Let us consider $B:\{1,0\}^m \to \{1,0\}$ be the function relates to Boolean monotone for every span function, i.e..... $(a_1, a_2, \ldots\ldots, a_n) \in \{1,0\}^m$, the following function should satisfy all the labeled functions.

$$b(a_1, a_2, \ldots\ldots, a_n) = 1 \Leftrightarrow \exists v \in \mathbb{R}^{1\times l} : v\,N = [0,1,1,\ldots 1] \& (\forall_i : a_{x(i)} = 0 \Rightarrow v_i = 0 \tag{1}$$

Here $b(a_1, a_2, \ldots\ldots, a_n) = 1$ is the span function indexed with a span vector $\left\{i \mid a_{x(i)} = 1\right\}$ & a span program constructed in Boolean retrieval functions.

b) Attribute-based Encryption

Attribute-based Encryption is explored with multiplier functions. The basic scenario of ABE is described as follows:

Initialization of system: Identify the prime number p, generator g of the group of generative functions G_O&G_T with the order of q, and then map the function $e: G_0 \times G_0^i \to G_T$, which is associated with a hash function $H:\{1,0\}^* \to G_0$. This hash function is combined with different attributes Lj, disjoint connection $\left(L_i \cap L_j = \phi\, for (i \neq j)\right)$, then the secret key of generative function is

$$SK \mid j \mid = \{x_i, b_i, i \in L_j\} \tag{2}$$

From the secret key, generate the public key from sources

$$PK[j] = \left\{e(g,g)^{x_i}, g^{b_i}, i \in L_j\right\} \tag{3}$$

Distribution of key generation: User v receives content from the set of attributes I (v, j), then associated secret key sk for each user attribute $i \in I[j,v]$, then hash-based secret key generation is

$$sk_{v,i} = g^{x_i} \mathrm{H}(v)^{b_i} \tag{4}$$

$x_i, b_i \in SK[j]$ be the secure user delivery of different public keys. Decryption is also performed using the secret key.

Data encryption with sender: Use the encrypt function in attribute-based Encryption $ABE.Enc(msg, \chi)$; the sender sends access tree structure encrypted message (cipher text(ct)) as follows:

$$CT = \left\langle R, \pi, ct_0 \left\{ct_{0,a}, ct_{1,a}, ct_{2,a}, \forall_a\right\}\right\rangle \tag{5}$$

Here $\pi(a)$ be the mapping connection R_a corresponding matrix with different attributes located and associated with the access tree of data.

The decryption of data by receiver: Use decryption function, i.e. $ABE.Dec\left\langle CT, \{sk_{i,v}\}\right\rangle$, CT be the encrypted text, receiver Vu explores cipher text CT with secret key sharing and then obtain output decrypted message with following conditions, i.e.

a) For each attribute $a \in A^{'}, dec(a) = \dfrac{C_{1,a} e(H(v), C_{3,a})}{e(sk_{\pi(a)}, v, C_{2,a})}$

b) V_u evaluates $msg = C_0 / \Pi_{a \in A^{'}} dec(a)$

3.1 Implementation and construction of NACPPA

a) Implementation procedure:

This section describes the implementation of security in NACPPA with the following calculation methods:

I. **Setup of System:** This calculation method is evaluated by the authority of a trusted user and organized by a third-party trusted user. This setup does not contain any input other than security parameters (in the form of attributes), i.e., α, β which are related to global public parameters, i.e., GPK (global public key)

II. **Generation of Key:** let us assume $\left(attri\left\{ucpk_{id,k} \mid k \in \ \right\}, \left\{KeyVerif_k \mid k \in \ \right\}, AMSK_d, GPK\right)$ the key generation method for attribute authority is evaluated by AA_d. It takes input as following attributes, i.e., attri, center-user-public-key ($ucpk_{i,k}$), key relates to verification VK_k, master key MK_d relates to public global parameters i.e. GPK. It gives output as secret_key SK_{ud} of the data user (DU), and the pair relates to the secret key, i.e., $\left(pk_o, sk_o\right)$ of the data owner (DO).

III. **Encryption of Files:** Domain authority evaluates and performs this encryption procedure. It takes input as plain text m, GPK, the secret key of DO's, public of DU's, i.e., pk_u, and converted cipher text CT_{fog_comm}. It gives all output as cipher text CT_1, CT_2 and stores that cipher text into the main cipher text server, i.e., cloud service provider.

IV. **GenTrap:** input as $\left(\omega, SK_{ud}, PK_O\right)$ Trap door generator calculation method is evaluated by the data user, i.e., DU, input for this calculation is presented with the search word ω, secret key SK_{ud}, and key with public PK_O then it generates output as Trapdoor (tw) & key relates to re-encryption value SK_{ud}^{w}

V. **The Decryption of Files:** This decryption calculation method evaluates data user and domain authority. It takes input as $\left(C, \Omega, RK\right)$, i.e., cipher text, which consists of partial decrypted text and key retrieval RK; plain text is the output for this calculation method.

b) Proposed NACPPA Schema

This section describes the basic algorithm implementation procedure of NACPPA, which is an extension to attribute set-based encryption to evaluate and handle the dynamic hierarchal structure of different users, as illustrated in Figure 2. Recall the procedure of domain authority and subordinate domain authorities and other data users concerning corresponding consumers and users related to data. In our proposed approach, the authority of trusted users evaluates privacy-related master key-assisted parameters present in top-level scenarios. Domain authority generates and distributes secure keys to subdomain authorities and users present at the next associated level. In our implemented system, each user contains a key structure generated by attributes with decrypted keys of users.

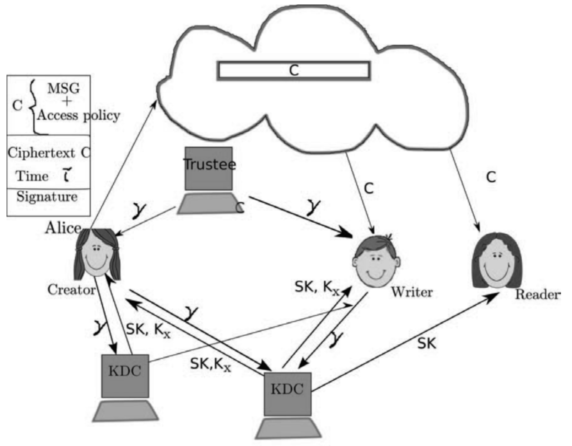

Figure 2. The User is represented in a hierarchal structure.

We describe the basic properties of NACPPA, i.e., it consists of the following methods:

System Setup: The setup calculation method is executed by the authority of domain for the creation of key_ public (PK) and key_ master (MK), d is the length of key_structure, by using bilinear map connection C be the prime order p with gen c and then evaluates random parameters $\alpha, \beta \in Z_p, \forall i(1,2)$. Then, the generation of both master and public keys based on the length of key d is

$$PK = \left(\mathbb{C}, c, h_1 = c^{\beta_1}, f_1 = c^{\frac{1}{\beta_1}}, h_2 = c^{\beta_2}, f_2 = c^{\frac{1}{\beta_2}}, e(c,c)^{\alpha}\right) \tag{6}$$

$$MK = (\beta_1, \beta_2, c^{\alpha}) \tag{7}$$

Creation and grant of domain authority: Domain authority is organized and associated with recursive attribute relations, i.e., $\left\{X_0, X_1, ..., X_m\right\}, X_i = \left\{x_{i,1}, x_{i,2}, ..., x_{i,n}\right\}$ with $a_{i,j}$, and related attributes. Creates key for domain authority, select identity no. of keys $r^{\{v\}}$ for domain authority. Random selection of identity keys for each user $a_{i,j}, 0 \le i \le m, 1 \le j \le n_i$ is utilized in the authority of the domain

$$DA(MK) = \begin{pmatrix} A, D = c^{\frac{\alpha + r^{\{u\}}}{\beta_1}}, D_{i,j} = c^{r_i^{(u)}}.H(a_{i,j})^{r_{i,j}^{\{u\}}}, \\ D_{i,j}^{'} = c^{r_{i,j}^{\{u\}}} for(0 \le i \le m, 1 \le j \le n_i), \\ E_i = c^{\frac{r^{\{u\}} + r_i^{\{u\}}}{\beta_2}} for(1 \le i \le m) \end{pmatrix} \tag{8}$$

In the above generation of domain authority master key, where E_i is the form of translation, unique rule formation $r^{\{u\}}$ relates to the attribute set A_i to $r^{\{u\}}$ with associative translating elements E_i & $E_i^{'}$ can be used as $E_i / E_i^{'}$ to translator to unique key generations; these details are used decryption calculation method again.

User grant/ selection of novel domain authority: In this scenario, a novel user "u" and subordinate with respect to authority of a domain and it is denoted as DA++, DA joined into the cloud system then DA verifies each user then DA generates key structure to every user and user gives grant to other relative users using authorizes which are derived from domain authority.

User Creation (DAMK, u$\aleph$): This calculation method uses the master key, which is generated by the DA with the structure of the key $\aleph$, then evaluates the key structure for a newly generated user $\dddot{\aleph}$, which is the combined key set structure of $\aleph$. Based on unique identifier sequences present in the DA master key, evaluates the secret key for the user described in the following equation:

$$(MK_{i+1}) = \begin{pmatrix} \dddot{\aleph}, \dddot{D} = D.f_i^{\dddot{r}^{\{u\}}}, \dddot{D}_{i,j} = D_{i,j}.c^{\dddot{r}_i^{(u)}}.H(a_{i,j})^{\dddot{r}_{i,j}^{\{u\}}}, \\ \dddot{D}_{i,j}' = D_{i,j}^{1}.c^{\dddot{r}_{i,j}^{\{u\}}} for(a_{i,j} \in \dddot{\aleph}), \\ \dddot{E}_i = E.f_2^{\dddot{r}^{\{u\}} + \dddot{r}_i^{\{u\}}} for(\aleph_i \in \dddot{\aleph}) \end{pmatrix} \tag{9}$$

MK$_{i+1}$ is the secret key of the user structure $\dddot{\aleph}$, and the recipient key is directly removed from the authority of the trusted user.

Encryption of File (PK, m, τ): plain message m is to be encrypted, M be the DEK file, τ be the tree access structure. The encryption calculation method is similar to attribute set-based Encryption but only supports polynomial equations q_a with access tree structure τ collected from randomly selected trusted authority from root domain authority. The encryption calculation method is described as follows:

$$Cipher\_T\,ext(CT) = \left(\begin{array}{c} \tau, \tilde{G} = M.e(c,c)^{\alpha.s}.G = h_1^s, G = h_2^s, \forall b \in B : \square \\ G_b = c^{q_b(0)}, G_b' = H(attr(b)^{q_b(0)}), \\ \forall_a \in A : \widehat{G}_a = h_2^{q_a(0)} \end{array}\right) \tag{10}$$

Where B defines the set of parent node with sub-leaf user in τ , and A is the set of the access tree structure τ .

Revocation of User: Any user who has access to owner-shared data withdrawn from the cloud system cannot access it from any location. We address this issue in our solution by using a re-encryption technique to access the shared file association where users' access is tied to the revoked format. The attributes of attribute set-based encryption are expanded by NACPPA to provide user revocation. If data owners share shared files, produce new keys for revoked users based on the domain authority security rights procedure.

File access operations: whenever the user sends the request to the cloud server, then the cloud server sends the encrypted request to the user, and then the user decrypts the updated data using $Dec(CT, SK_u)$ the decryption procedure as follows.

The decryption of Files: This calculation method takes input as cipher text and key structure. Firstly, the decryption procedure verifies user key structure k concerning associative access tree structure τ and cipher text content and is accessed from the data owner. Satisfy all the conditions with τ and key structure of the user "u" and then decrypt the entire content; if not satisfy the conditions, then evaluate/perform decryption_method. The decryption method is described as follows:

$$Decr(CT, SK_u, i, t) = e\left(D_{i,j}, G_t\right) / \left(D'_{i,j}, G'_t\right) = e(c,c)^{r_i^{\{u\}}}.q_t(0) \tag{11}$$

The decryption method for all stored encrypted content with translated polynomial interpretation F$_z$ is described as follows:

$$F_{z)} = e\left(\hat{G}_z, E_i / E_{i'}\right).F'_z = e\left(c,c\right)^{r^{\{u\}}}.q_z(0) \tag{12}$$

Based on the above decryption procedure, the message to be evaluated as $M = \hat{G}.F / e\left(G, D\right)$.

4. EXPERIMENTAL EVALUATION

To calculate the efficiency of the proposed approach to design an empirically secure cloud with users accessed by multiple files from different data owners in the cloud, we have implemented the NACPPA framework based on the working procedure of ABE. Using the latest version related to CloudSim, Java, and Netbeans are used to set up the newest cloud environment. Each host consists of 2.4 Hz with 4-8 GB RAM and 1TB of data storage for this implementation. Using these requirements, our proposed approach explores the following sequences:

Setup_NACPPA: It generates both public and master keys, i.e., key_ public (PK) and key_ master (MK) assumptions
NACPPA_keyGen: Implemented PK and MK to generate a key related to private operations with critical structures. Actual structure depth supports 1 or 2 support functions.
NACPPA_keyDeleg: Based on PK and MK, which are related to the authority of the domain, this delegates some methods of DA's private keys for the newly generated user. In domain authority, the delegated key is used for the private key.
NACPPA_enc: Based on access tree policy conditions, generates encrypted file using PK.
NACPPA_Dec: Using private, decrypt the files
NACPPA_rec: Using PK, encrypt all the files using the private key and generate re-encryption for both encrypted and decrypted files. Note that generated private is used to decrypt the file using encrypt file operations.

Experimental results of the implemented approach concerning the time taken by different operations with different methods are to be calculated. The following figures show the time taken for other instances during authentication in the client cloud with the proposed approach. To assess the effectiveness of the proposed NACPPA approach with a comparison of different authentication approaches like ASBE and CP-ABE, the following results (generated in the proposed approach) give efficient security response time, encryption for files, decryption for files, and access tree generation time for different users. The results also provide average accuracy with memory utilization for user operations like upload, requests, and download requests for efficient and secure data storage in a distributed environment. Table 1 describes total time values for processing different user instances.

Table 1. Total time values for different user instances

Different users	CP-ABE	KP-ABE	ASBE	NACPPA
10	5.2	3.8	4.7	2.9
30	6.3	5.9	6.3	4.4
50	7.5	6.5	5.6	4.8
70	8.4	7.3	8.2	5.7
100	9.5	8.4	7.3	6.4

As shown in Table 1 and Figure 3, compared to traditional approaches, i.e., ASBE, KP-ABE took approximately equal time to explore user instances on the cloud whenever user instances increased, and those approaches took more time to execute the services of different users. Compared to the proposed method, it took less time than existing approaches.

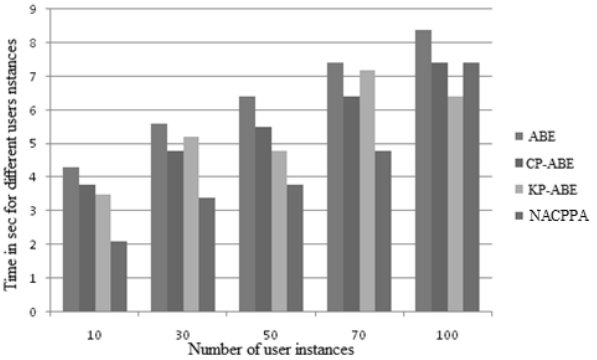

Figure 3. Performance evaluation for cloud setup environment to all the user operations.

Encryption time for different user instance requests with the upload of original content in the secure format in the cloud with different values is described in Table 2.

Table 3. Encryption time values for different user instances

Different users	CP-ABE	KP-ABE	ASBE	NACPPA
100	5.3	4.7	4.6	4.5
200	6.4	5.8	6.3	3.1
300	7.4	7.7	6.3	3.3
400	8.3	7.2	8.2	4.7
500	9.6	8.4	9.4	7.6

Table 2 and Figure 4 show the encryption time evaluation values and performance evaluation of different approaches in encryption. ASBE and CP-ABE took more time to increase the user instance concerning other services. The proposed method took less time to encrypt files uploaded by different users.

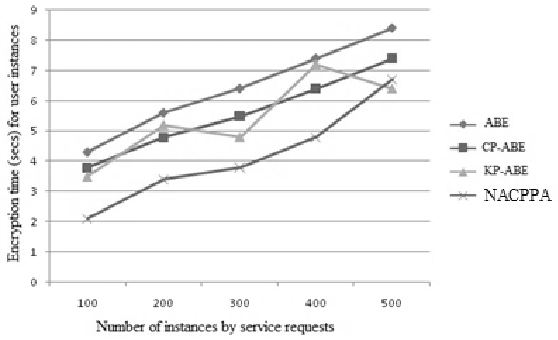

Figure 4. Performance evaluation of different approaches with encryption time.

Different users want to download shared files from users of data owners to evaluate the decryption time values, shown in Table 3 in secure cloud storage.

Table 3. Description time values

No. of User Instances	CP-ABE	KP-ABE	ASBE	NACPPA
100	4.8	5.8	5.3	4.7
200	5.3	6.7	5.9	3.7
300	4.8	8.5	6.4	3.3
400	7.4	6.9	5.7	3.9
500	6.8	7.5	7.9	6.4

Table 3 and Figure 5 show the decryption time evaluation values and performance evaluation of different approaches in decryption. Also, ASBE, KP-ABE, and CP-ABE took more time to increase the user instance concerning other services. The proposed method took less time to decrypt files uploaded by different users with different instant services.

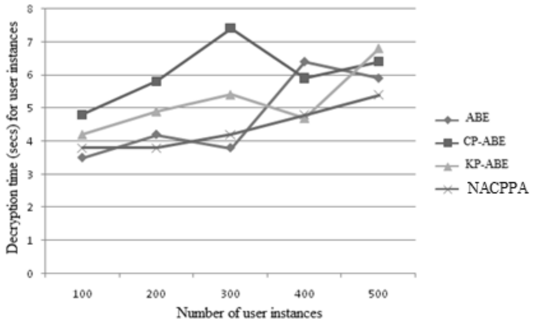

Figure 5. Performance evaluation of decryption time of different approaches.

Table 4 and Figure 6 show the performance evaluation of different approaches in memory utilization to explore user instance services.

Table 4. Utilization of memory values in processing user operations in secure cloud storage.

Users	CP-ABE	KP-ABE	ASBE	NACPPA
100	4652	3642	4887	4760
200	5327	4326	5226	5626
300	7356	6974	4745	4026
400	22132	5796	6354	5356
500	24553	8964	6785	4324

The utilization of memory with different user operations, like storing data securely and accessing tree structures with feasible secure storage, was described in Table 4.

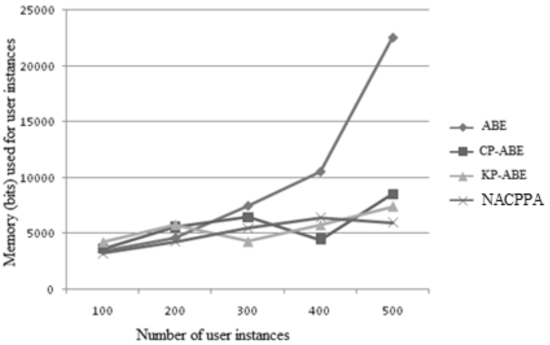

Figure 6. Performance evaluation of the proposed approach with traditional approaches in terms of memory.

ASBE and KP-ABE took a lot of memory whenever increasing the user instance services. Because of less time complexity, the proposed approach runs with the lowest memory utilization in processing users' services concerning other systems.

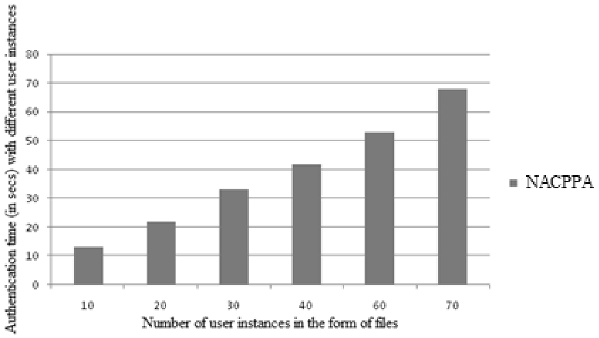

Figure 7. Performance evaluation of generation of access tree structure.

Figure 7 shows the performance evaluation of the access tree structure concerning different users' tree construction to store data securely. Table 5 shows the average precision of accuracy values of the proposed approach with different user instance files stored in a secure format in the cloud. Based on the above results, if we increase other user instances, traditional methods give less accurate results when compared to the proposed approach derived from secure cloud storage.

Table 5. Average accuracy values for different user instances.

Users	CP-ABE	KP-ABE	ASBE	NACPPA
100	4.5	5.9	5.1	3.6
200	5.3	6.6	5.8	4.4
300	4.8	6.3	6.3	3.5
400	7.4	6.8	5.4	4.9
500	6.9	7.2	7.6	5.4

Support different user instances in exploring data from a secure cloud, as described in Table 5.

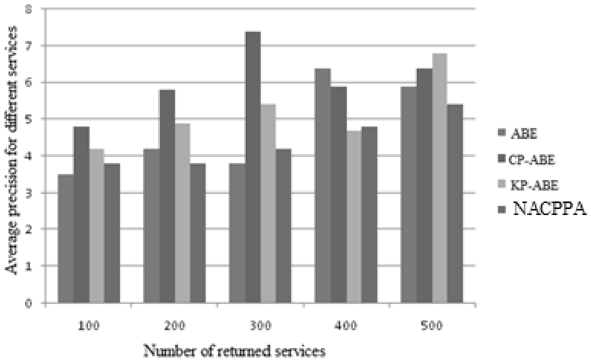

Figure 8. Performance of accuracy with different user secure operations.

Figure 8 shows the accuracy of different users in performing data security operations in a distributed environment. Figures 3-8 show the performance of total user instances, time for Encryption, and decryption concerning memory utilization. The proposed approach performs efficiently compared to conventional methods like ASBE, KP-ABE, and CP-ABE designed with multi-file sharing cloud environments.

4. CONCLUSION

In this report, we carried out a novel secure authentication approach, i.e., NACPPA, to provide green, scalable, flexible user supply get admission to manage shape in cloud computing. NACPPA virtually put into effect a hierarchal structure for accessing a person's files by applying a facts delegation manner, which is present in characteristic set-based Encryption. NACPPA does not best support user protection, and it achieves the advanced idea, i.e., revocation of consumers in statistics sharing if more than one project attribute is a gift. We speak about the security performance procedure of NACPPA with specific idea-level calculation techniques. The applied experiments indicate green, comfy overall performance evaluation and evaluation of advanced safety efficiencies in cloud computing. Further extension for this is to control the corporation's keys and discuss how they may help multi-person relaxed statistics sharing depending on a cloud server in cloud computing.

References:

Alturi, V., & Ferraiolo, D. (2011). Role-based access control. In H. C. A. van Tilborg & S. Jajodia (Eds.), *Encyclopedia of cryptography and security* (pp. 829–833). Springer. https://doi.org/10.1007/978-1-4419-5906-5_829

Chen, Y., Sun, W., Zhang, N., Zheng, Q., Lou, W., & Hou, Y. T. (2019). Secure remote monitoring framework supporting efficient fine-grained access control and data processing in IoT. *IACR Cryptology ePrint Archive, 2019*, 86.

Helil, N., & Rahman, K. (2017). CP-ABE access control scheme for sensitive data set constraint with hidden access policy and constraint policy. *Security and Communication Networks*, *2017*, 1–13. https://doi.org/10.1155/2017/2713595

Huang, Q., Yang, Y., & Wang, L. (2017). Secure data access control with ciphertext update and computation outsourcing in fog computing for Internet of Things. *IEEE Access, 5*, 12941–12950. https://doi.org/10.1109/ACCESS.2017.2727054

Li, J., Lin, X., Zhang, Y., & Han, J. (2017). KSF-OABE: Outsourced attribute-based encryption with keyword search function for cloud storage. *IEEE Transactions on Services Computing, 10*(5), 715–725. https://doi.org/10.1109/TSC.2015.2399311

Liu, Q., Wang, G., & Wu, J. (2014). Time-based proxy re-encryption scheme for secure data sharing in a cloud environment. *Information Sciences, 258*, 355–370. https://doi.org/10.1016/j.ins.2012.09.034

Rasori, M., Perazzo, P., Dini, G., & Yu, S. (2022). Indirect revocable KP-ABE with revocation undoing resistance. *IEEE Transactions on Services Computing, 15*(5), 2854–2868. https://doi.org/10.1109/TSC.2021.3071859

Roslin Dayana, K., & Shobha Rani, P. (2023). Trust aware cryptographic role-based access control scheme for secure cloud data storage. *Automatika, 64*(4), 1072–1079. https://doi.org/10.1080/00051144.2023.2243144

Ruj, S., Stojmenovic, M., & Nayak, A. (2014). Decentralized access control with anonymous authentication of data stored in clouds. *IEEE Transactions on Parallel and Distributed Systems, 25*(2), 384–394. https://doi.org/10.1109/TPDS.2013.38

Sahai, A., & Waters, B. (2005). Fuzzy identity based encryption. In R. Cramer (Ed.), *Advances in cryptology – EUROCRYPT 2005* (Vol. 3494, pp. 457–473). Springer. https://doi.org/10.1007/11426639_27

Samanthula, B. K., Elmehdwi, Y., Howser, G., & Madria, S. (2015). A secure data sharing and query processing framework via a federation of cloud computing. *Information Systems, 48*, 196–212. https://doi.org/10.1016/j.is.2013.08.004

Wang, G., Liu, Q., Wu, J., & Guo, M. (2011). Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. *Computers & Security, 30*(5), 320–331. https://doi.org/10.1016/j.cose.2011.05.006

Xu, S., Yang, G., Mu, Y., & Deng, R. H. (2018). Secure fine-grained access control and data sharing for dynamic groups in the cloud. *IEEE Transactions on Information Forensics and Security, 13*(8), 2101–2113. https://doi.org/10.1109/TIFS.2018.2810065

Zhao, Z., & Wan, J. (2017). Verifiable outsourced ciphertext-policy attribute-based encryption for mobile cloud computing. *KSII Transactions on Internet and Information Systems, 11*(6), 3254–3272. https://doi.org/10.3837/tiis.2017.06.024

N. Praneetha
KLEF University,
Vaddeswaram- 522 302,
India
nyshpranee@gmail.com
ORCID 0000-0002-7027-9516

S. Srinivasa Rao
KLEF University,
Vaddeswaram-522 302,
India
srinu1479cse@kluniversity.in
ORCID 0000-0002-6183-7088

Maheswara Rao V V R
Shri Vishnu Engineering
College for Women,
Bhimavaram, India
mahesh_vvr@yahoo.com
ORCID 0000-0002-0503-7211

I S Siva Rao
GITAM Deemed to be University
Visakhapatnam,
India
isro75@gmail.com
ORCID 0000-0001-9181-7507

P. S. Brahmanandam
Shri Vishnu Engineering College
for Women,
Bhimavaram- 534202,
India
dranandpotula@svecw.edu.in
ORCID 0000-0001-9777-3496