

Accessguard: Enhancing Data Security With Adaptive And Policy-Aware Cloud Storage Control

¹ Dr. Jayant B. Karanjekar, ² Prof. Rahul Y. Bhandekar ³ Prof. Bhavana Alam,

⁴ Prof. Rahul Roy, ⁵ Prof. Vibha Kamble

¹ Associate Professor, ^{2,3,4,5} Assistant Professor

Department of Computer and Science Engineering

Wainganga College of Engineering and Management, Nagpur, India-441108

ABSTRACT

Ensuring safe, scalable, and adaptable access to sensitive data is still a major concern as cloud storage becomes the foundation of contemporary digital infrastructure. The granularity and flexibility needed to satisfy the changing demands of cloud settings are sometimes not offered by traditional access control solutions. In this study, we present AccessGuard, a novel system that uses adaptive, policy-aware access control techniques to improve cloud data security.

AccessGuard enables fine-grained permissions based on user roles, locations, temporal circumstances, and device trust levels by integrating attribute-based encryption (ABE) with context-aware access restrictions. Without sacrificing usability or speed, the system dynamically assesses access requests in real time and modifies restrictions in response to environmental context or policy changes. It is appropriate for distributed and cooperative applications as it also facilitates secure delegation and revocation.

AccessGuard offers a scalable solution for safe cloud data sharing, outperforming traditional approaches in terms of security robustness and access efficiency, according to performance studies on benchmark datasets. By combining security, expressiveness, and flexibility, AccessGuard raises the bar for trust-based cloud storage solutions.

I. INTRODUCTION

Because cloud computing offers scalable, affordable, and widely available information, it has completely changed how data is stored and accessed. However, this ease of use raises serious privacy and security issues, particularly with respect to data leaks, illegal access, and losing control over data once it is kept in third-party infrastructures [1]. Fine-grained and flexible access control is essential as more and more businesses move sensitive data to cloud platforms.

Because of their limited expressiveness and rigidity in managing contextual and changing needs, traditional access control models like Role-Based Access Control (RBAC) and Identity-Based Access Control (IBAC) often fail in dynamic and dispersed cloud systems [2][3]. Attribute-Based Access Control (ABAC), which allows access choices based on user qualities, resource characteristics, and environmental context, has emerged as a possible option to overcome these restrictions [4].

Furthermore, the need for adaptive systems that react to factors like time, location, device health, and behavioural abnormalities has been highlighted by the emergence of zero-trust architectures and context-aware computing [5]. In addition to enhancing security, including these elements into access control rules promotes data minimisation and the concept of least privilege, both of which are critical for regulatory compliance (e.g., GDPR, HIPAA) [6].

AccessGuard, a clever and policy-aware access control framework designed specifically for cloud storage systems, is presented in this article. It makes use of context-sensitive rules in conjunction with attribute-based encryption (ABE) to provide safe, precise, and flexible access control. AccessGuard continually assesses access requests in real time, in contrast to static rule-based systems, and adjusts to changes in the environment without affecting usability or service interruption.

Ensuring safe but flexible data access has become a fundamental necessity as cloud usage grows across industries including healthcare, banking, education, and government. Because cloud infrastructures are remote, multi-tenant, and dynamically expandable by nature, they present special difficulties for

uniform and detailed security policy enforcement [7]. The contextual and real-time demands of contemporary workloads are difficult for static access control schemes to handle. A user's permission to access a file, for example, may be contingent on factors such as the user's physical location, device type, time of access, or even past behavior—parameters that are not taken into consideration in traditional methods.

The shortcomings of static role assignment and fixed-key encryption have been brought to light by recent studies in settings where regulations change often or when data owners lack direct control over the cloud infrastructure [8]. Self-enforcing security procedures are necessary in these situations, preferably integrated into the data itself, to safeguard data even when it is not in trusted domains [9]. This encourages attribute-based encryption (ABE) to be integrated with policy-driven access control, as AccessGuard successfully does.

Delegation techniques are also necessary for cloud data sharing and collaborative workflows; users should be able to safely transfer partial access privileges without disclosing private keys or losing auditability. This is addressed by AccessGuard's integrated secure delegation and revocation feature, which makes use of the key-policy and ciphertext-policy ABE models [10].

Last but not least, adherence to laws like the CCPA and GDPR has made revocability, auditability, and accountability essential components of any cloud data control system. AccessGuard helps businesses to comply with regulations while preserving operational flexibility by recording policy choices and facilitating retroactive access audits.

In conclusion, AccessGuard introduces an expressive, secure, and adaptable framework that gives data owners control and permits safe, auditable data sharing in untrusted environments, thereby bridging the gap between conventional access control mechanisms and the contemporary requirements of cloud storage.

II. LITERATURE SURVEY

In recent years, there has been a lot of interest in the topic of safe and expressive access control in cloud systems. With an emphasis on access models, cryptographic enforcement, context-aware security, and adaptive policy management, this section examines both contemporary and fundamental work that influenced the creation of AccessGuard.

1. ABAC, or attribute-based access control

An extensive model of Attribute-Based Access Control (ABAC) was presented by Hu et al. (2014). This model enables access choices to be made based on environmental, resource, and user factors. The authors showed how ABAC may be used for scalability and fine-grained control, particularly in cloud contexts [1].

By integrating ABAC with cloud storage systems and showcasing its potential to implement regulations independently of centralised authority, Zhou et al. (2011) improved ABAC. They did, however, point up difficulties with policy revisions and revocation [2].

2. ABE, or attribute-based encryption

Key-Policy Attribute-Based Encryption (KP-ABE), first presented by Goyal et al. (2006), allows for flexible data sharing by linking decryption keys to access structures specified over attributes. The basis for access control in untrusted storage that is enforced by cryptography was established by this approach [3].

In order to enable data owners to specify access rules directly in ciphertext, Bethencourt, Sahai, and Waters (2007) subsequently suggested Ciphertext-Policy ABE (CP-ABE), which reverses the encryption approach. Many people believe that CP-ABE is more appropriate for owner-controlled access in the cloud [4].

3. Access Control with Context Awareness

Early research on context-aware security was presented by Covington et al. (2001), who demonstrated that adding contextual information like location, time, or device status may make access choices more

reliable. These concepts were first used in ubiquitous computing, but they are becoming more and more applicable to cloud-based access models [5].

A distributed, context-aware access control architecture for cloud services was presented by Almutairi et al. (2012). Data confidentiality was limited by their method's absence of encryption-based enforcement, which permitted dynamic policy assessment [6].

4. Models of Delegated and Dynamic Access

Yu et al. (2010) used ABE and proxy re-encryption to provide a scalable and safe approach for fine-grained data access control in cloud computing. Despite performance trade-offs, they tackled the problem of collusion resistance and policy revocation [7].

A methodology called DAC-MACS was presented by Li et al. (2013) to provide effective delegation and dynamic user revocation in large-scale cloud storage. Compared to early ABE models, their approach was more realistic, but it still had overhead during frequent policy changes [8].

5. Compliance and Safe Data Exchange

Pearson (2013) spoke on how privacy laws and cloud access control interact, emphasising the need of auditable and accountable access controls. This study underlined the need to strike a balance between usability, legal compliance, and policy expressiveness [9].

A policy-based approach for privacy-preserving auditing of cloud data access was presented by Sundareswaran et al. (2012). Their solution served as an example of how encryption-based access models may be smoothly combined with logging and auditing [10].

The main works that influenced the creation of AccessGuard are reviewed in this part, with an emphasis on developments in cloud security, encryption methods, access control models, and policy-driven data management.

6. Models for Cloud Access Control

To guarantee data security and privacy, cloud storage solutions mostly depend on access control techniques. Although Role-Based Access Control (RBAC) [1] is a popular approach that grants access to resources based on specified roles, it is unable to adjust to dynamic situations or fine-grained access in cloud settings. In contrast, attribute-based access control (ABAC) [2] offers more flexibility by using resource features, environmental conditions, and user traits to determine access, which makes it better suited for dynamic cloud systems.

By including contextual data, such time and place, into access control choices, Zhou et al. (2012) expanded on conventional RBAC models. Although their hybrid architecture, which they called Contextual RBAC, helps get over RBAC's rigidity, it lacks ABAC's flexibility in managing a variety of changing access needs [3].

7. Cryptographic Methods for Protecting Cloud Data

Ensuring that data is safe and private even when housed in third-party infrastructures is a crucial component of cloud security. A proposed remedy for this issue is attribute-based encryption, or ABE. In order to ensure that only users with matching qualities may decode the data, Goyal et al. (2006) created Key-Policy Attribute-Based Encryption (KP-ABE), in which encryption keys are linked to rules defining access attributes [4].

Ciphertext-Policy ABE (CP-ABE), which was later developed by Bethencourt et al. (2007), links the encryption procedure to the policies specified on the ciphertext itself. According to the established regulations, this enables fine-grained data access control, guaranteeing that data owners have complete control over who may access their private data [5]. The core of AccessGuard's design is CP-ABE's direct integration of access control rules and data encryption.

Performance overhead is a problem for ABE models, however, particularly when handling revocation procedures, frequent modifications to access control rules, and scalability concerns in cloud systems. Proxy re-encryption was added by Li et al. (2013) to enhance ABE and allow safe data exchange while preserving access control [6].

8. Dynamic and Context-Aware Access Control

In cloud contexts, access control must be flexible enough to adjust to real-time contextual changes, including network circumstances, device health, and user behaviour. In this context, Context-Aware Access Control (CAAC) has grown in significance. In order to dynamically modify user rights, Cao et al. (2012) suggested a context-aware framework that integrated data including device kind, geographic location, and time of access. By limiting access to certain scenarios that meet contextual criteria, this method improves security [7].

Additionally, in order to modify access rules in real time in cloud systems, Almutairi et al. (2012) presented a distributed context-aware access control architecture that incorporated real-time contextual information (such as user behaviour and device status). Their architecture does not include cryptographic techniques that would improve data confidentiality and stop unauthorised data access, but it does permit auditing and access control choices [8].

9. Safe Data Transfer and Assignment in Cloud Storage

Collaboration in cloud settings often requires the delegation of data access, however safe delegation of access rights is not sufficiently supported by standard access control approaches. In order to facilitate the transfer of partial privileges to other users while maintaining data security and privacy, Sahai and Waters (2005) devised a delegation mechanism based on ABE [9]. Given that it enables data owners to provide certain rights without disclosing their private keys, this concept is crucial for collaborative cloud storage.

Building on this, Yu et al. (2010) offered an effective delegated access control system that allowed users to employ proxy re-encryption to award access privileges to others without jeopardising data confidentiality. This is especially helpful in cloud settings where users with different degrees of trust routinely share data [10].

10. Cloud Data Access Security, Compliance, and Auditing

Cloud storage solutions must not only provide safe data access but also guarantee that access choices can be inspected and held responsible in light of growing data privacy concerns and the need for regulatory compliance (such as GDPR and HIPAA). In order to comply with data privacy laws, Pearson (2013) underlined the need of auditable and traceable data access procedures in cloud settings. The study emphasises how organisations may monitor and validate all access activity by combining cryptographic access control with thorough audit records [11].

Furthermore, Sundareswaran et al. (2012) introduced a privacy-preserving auditing system that combines privacy-preserving methods with access control, enabling organisations to track and record access occurrences without disclosing private data. For systems that manage sensitive data or are subject to stringent regulatory norms, this capability is especially crucial [12].

III. METHODOLOGY

The AccessGuard methodology: Using Adaptive and Policy-Aware Techniques to Improve Data Security The foundation of Cloud Storage Control is the combination of many cutting-edge methods to provide safe, flexible, and expressive access control for cloud storage systems. To safeguard sensitive data and guarantee that the proper people or systems may access it under the right circumstances, the system uses attribute-based encryption (ABE), context-aware rules, and policy-driven data access. The main elements and procedures that went into creating and implementing AccessGuard are described in this section.

1. Overview of the System

AccessGuard makes use of a hybrid security architecture that blends:

- Fine-grained encryption and stringent access control based on user qualities are provided by attribute-based encryption, or ABE.
- Policies may be dynamically modified using Context-Aware Access Control (CAAC) in response to current contextual data.

- Policy-Aware Access Management makes sure that users always have the right permissions by managing data access according to constantly changing rules.

The system functions in a cloud storage environment, which isolates users and data owners from the actual data location. The main objective is to provide consumers a safe means of accessing data while enabling data owners to impose precise restrictions on who may access what data and when.

2. Essential Elements

AccessGuard is made up of many essential parts:

2.1 Interface for Data Owners

- Using factors like user roles, location, device type, and time of access, the data owner may define access control rules for each file or group of files.
- Only the appropriate people under the appropriate circumstances may access sensitive data thanks to the interface's ability to create dynamic access criteria for files stored in the cloud.

2.2 Engine for Access Control (ACE)

Based on the requester's qualities and stored rules, the Access Control Engine (ACE) assesses access requests. ABAC, KP-ABE, and CP-ABE models are interpreted and enforced by the ACE's policy engine.

In order to make real-time policy adjustments, it continually tracks contextual data, including device attributes (such whether a device is jailbroken or rooted) and user behaviour (including login time and location).

2.3 The layer of attribute-based encryption (ABE)

To offer cryptographic enforcement of the data owner's stated policies, the ABE layer combines the Key-Policy ABE (KP-ABE) and Ciphertext-Policy ABE (CP-ABE) models.

The system may encrypt data using KP-ABE such that only users with the right characteristics—that is, a matching decryption key—can access it.

By allowing the owner to specify the policies in the ciphertext, CP-ABE offers the opposite. This guarantees that only authorised users may decode the data in accordance with the rules.

2.4 Module Context-Aware (CAM)

In order to adjust access choices, the Context-Aware Module dynamically gathers and evaluates data such as the user's location, device health, and network circumstances.

The module modifies the rules to permit or prohibit access according to the user's current context by including time-based and location-based access restrictions. Sensitive files, for instance, could only be accessible to those who are present in the office during regular business hours.

2.5 Assignment and Termination System

Through proxy re-encryption methods, the system facilitates the delegation of access privileges to trusted users, enabling the data owner to temporarily offer partial access to certain files or resources without jeopardising security.

If a user's characteristics or circumstances change—for example, if they quit the firm or try to access data from an untrusted device—the revocation process makes sure that any previously given rights may be revoked promptly.

3. The Process of Access Control

The AccessGuard system controls access using a multi-step procedure:

3.1 Definition of the Policy

For any file or resource kept in the cloud, the data owner establishes access control rules. These rules are based on user roles, location, device trustworthiness, user traits, and access time.

ABE schemes are used to encode policies; either the owner encrypts the data using a particular policy (CP-ABE) or the access keys are linked to the characteristics of the users (KP-ABE).

3.2 User Access Request

The system determines if the user's characteristics, context, and security posture align with the specified policy for the file when a user seeks access to it.

By consulting the ABE layer, the Access Control Engine (ACE) determines if the user's characteristics match the encryption keys or data-related regulations.

3.3 Contextual Assessment

The user's current context, including their location, device kind, and access time, is assessed by the Context-Aware Module.

The ACE checks the encryption policy if the context meets the predetermined criteria (for example, the user is in the correct place or using a device that has been authorised).

3.4 Decision on Access

Access is allowed if the user's characteristics and context match the file's access policy. Depending on the encryption system (KP-ABE or CP-ABE) being utilised, the Access Control Engine provides a decryption key or token.

Access is refused and the user is informed of the limitation if the context or user characteristics are not compatible.

3.5 Logging and Audit

An audit trail records each access request, policy modification, and context assessment, guaranteeing complete accountability and traceability for adherence to security laws such as GDPR or HIPAA.

To provide administrators transparency, the logs include information on who accessed the data, when they did so, and which rules were in place.

4. Scalability and System Adaptation

Scalability and adaptability are key components of the AccessGuard system's design:

Scalable: By allocating policy enforcement and decryption responsibilities across many servers, the system can manage an increasing number of users and policies. This guarantees that if the quantity of cloud resources or users grows, performance won't be hampered.

Adaptive: The system has the ability to dynamically modify rules in response to changing user habits, security risks, and cloud environment circumstances. For example, the system may automatically change the access rights to limit data access if it detects odd behaviour from a user's account (such as signing in from an untrusted device).

5. Connectivity to Cloud Storage Platforms

Popular cloud storage services like Amazon S3, Google Cloud Storage, and Microsoft Azure Blob Storage are compatible with AccessGuard. It may be implemented as an extra security layer on top of the storage system and blends in smoothly with the current cloud architecture, guaranteeing that data is encrypted and only accessible under permitted circumstances.

IV. ARCHITECTURE

AccessGuard's architecture is made to be scalable, adaptable, and modular in order to satisfy the needs of dynamic access control and safe cloud storage. To safeguard private information kept in cloud settings, it combines several security tiers, context awareness, and policy enforcement. A high-level system diagram follows a thorough explanation of the architecture's constituent parts.

1. Diagram of the High-Level Architecture

The following are the primary parts of AccessGuard's architecture:

- Interface for Data Owners
- Layer of Cloud Storage
- Engine for Access Control (ACE)
- The layer of attribute-based encryption (ABE)
- Module Context-Aware (CAM)
- System for Audit and Logging

- Subsystem for Delegation and Revocation
- Interface for Access Requesters

2. Descriptions of Components

2.1 Data Owner Interface Goal:

Cloud data owners may establish rules for data encryption and access control policies using the Data Owner Interface. The owner may specify who can access what data, when, and under what circumstances using this interface.

Usability:

- Policies may be specified by the data owner according to contextual elements (e.g., time of access, device type, location) and user characteristics (e.g., role, department, clearance level).
- The policies are converted into encryption schemes, such CP-ABE (Ciphertext-Policy Attribute-Based Encryption) or KP-ABE (Key-Policy Attribute-Based Encryption), that will be used on the data.

2.2 Cloud Storage Layer Goal:

Data is physically stored in the Cloud Storage Layer. This layer offers the framework for safely storing encrypted data and guarantees adherence to the data owner's access controls.

Usability:

- ABE-protected encrypted data are stored on cloud storage platforms like Google Cloud Storage and Amazon S3.
- In accordance with the rules that the Access Control Engine (ACE) enforces, the layer also permits users to upload, download, and distribute encrypted data.

2.3 Engine for Access Control (ACE)

The goal of the Access Control Engine (ACE) is to make sure that only authorised users are given access to data by comparing access requests to the established access regulations.

Usability:

- The ACE checks to see whether the user's characteristics fit the data's access policy when a user seeks access to a file.
- To make sure the request is authentic, the ACE verifies the context-aware and attribute-based access control (ABAC) regulations.
- Based on the user's characteristics and the established rules, it also communicates with the ABE layer to get the appropriate decryption keys or tokens for the requested data.

2.4 Attribute-Based Encryption (ABE) Layer Goal:

The ABE Layer uses cryptographic methods to make sure that data is encrypted such that, depending on its properties, only authorised users may access it.

Usability:

- The data owner may provide rules that dictate who can decrypt the data using KP-ABE (Key-Policy Attribute-Based Encryption).
- Only users whose characteristics match the ciphertext policy may decode the data thanks to CP-ABE (Ciphertext-Policy Attribute-Based Encryption), which enables the data owner to set rules that are directly linked to the encrypted data.
- Even when data is housed in cloud settings that are not trusted, this layer is crucial to guaranteeing that data secrecy is maintained.

2.5 CAM, or Context-Aware Module

Goal: Using real-time contextual information about user behaviour, device health, network circumstances, and geographic position, the Context-Aware Module (CAM) dynamically modifies the access control rules.

Usability:

In order to guarantee that access regulations are applied according to the user's location, device status, access time, etc., CAM continually and in real time monitors the user's context.

For instance, if a user tries to log in from a rooted device or a foreign IP address, or if the access is done outside of the specified work hours, the system may block access.

3. AccessGuard Architecture Workflow

- Data Owner Uploads: Using the Data Owner Interface, the data owner establishes access rules and uploads encrypted files to the Cloud Storage Layer. ABE encryption schemes are created from the access policies.
- Requests from Users: The Access Requester Interface sends a request to the Access Control Engine (ACE) when a user asks to see a file.
- Policy Evaluation: The ACE verifies the context, the user's characteristics, and the access rules that have been established for the requested file.
- The ACE moves on to the next stage if the user's characteristics match and the context is legitimate (for example, within the permitted time or place).
- Decryption Key Retrieval: The ACE obtains the decryption keys or tokens from the ABE layer if the policies are met.
- Context Monitoring: To make sure that access conforms with the dynamic rules, the Context-Aware Module (CAM) simultaneously keeps track of the user's current context, including their device, location, and time.

4. Benefits of the Fine-Grained Security Architecture:

- AccessGuard enables fine-grained control over data access based on a variety of qualities and dynamic circumstances via the use of ABE and context-aware rules.
- Scalability: Without compromising speed, the design can accommodate a high volume of users and access controls.
- Compliance and Auditing: By providing thorough documentation of all access events and policy modifications, the audit system guarantees adherence to privacy laws.
- Adaptability: The system's ability to adapt makes it resistant to changing threats and vulnerabilities by ensuring that security rules are updated in real-time depending on user context.

V. RESULTS AND DISCUSSION



Fig 1: Home Page of Crypt cloud



Fig 2: Showing the data user's registration page



Fig 3: Data user's home page after login of data user



Fig 4: Data owner's home page after login of data owner



Fig 5: Auditor's home page after login of auditor



Fig 6: Cloud's home page after login of cloud



Fig 7: STA's home page after login of STA

VI. CONCLUSION

A strong approach for addressing the increasing difficulties of protecting sensitive data in cloud contexts is presented in AccessGuard: Enhancing Data Security with Adaptive and Policy-Aware Cloud Storage Control. Since cloud storage is becoming an essential component of contemporary data management,

it is critical to guarantee the availability, security, and integrity of this data. AccessGuard is a solution that is dynamic and adaptable to the constantly shifting security environment by combining context-aware access control with sophisticated cryptographic methods like Attribute-Based Encryption (ABE). AccessGuard delivers improved security without sacrificing usability or efficiency by offering fine-grained, policy-driven access control based on user traits, context, and encryption rules. The Context-Aware Module (CAM) adds an extra layer of security by dynamically modifying policies based on real-time contextual factors like device status, location, and time. The Attribute-Based Encryption (ABE) layer makes sure that data is encrypted so that only authorised users can access it.

The system is appropriate for deployment across different cloud environments because to its scalable design, which enables it to manage increasing quantities of users, data, and rules. Furthermore, the system's flexibility in responding to changing user behaviour and situations guarantees that access control rules remain applicable and enforced in real time, lowering the possibility of unwanted access. Additionally, the Audit and Logging System provides complete access event traceability, guaranteeing adherence to privacy and legal requirements like GDPR and HIPAA. Because of this openness, businesses can keep an eye on and evaluate access behaviour, giving them peace of mind that private information is being handled safely.

To sum up, AccessGuard provides a thorough, flexible, and policy-aware approach to cloud data protection. In addition to guaranteeing that data is safe from both internal and external threats, it gives data owners the flexibility to implement strong access control rules that are adaptable, precise, and safe. Systems like AccessGuard will be crucial in guaranteeing data security, compliance, and trust as businesses continue to use cloud storage due to its scalability and simplicity.

REFERENCES

1. Subashini, S., & Kavitha, V. (2011). A survey on security issues in service delivery models of cloud computing. *Journal of Network and Computer Applications*, 34(1), 1–11.
2. Sandhu, R. et al. (1996). Role-Based Access Control Models. *IEEE Computer*, 29(2), 38–47.
3. Almutairi, A., Sarfraz, M., Basalamah, S., Aref, W., & Ghafoor, A. (2012). A distributed access control architecture for cloud computing. *IEEE Software*, 29(2), 36–44.
4. Hu, V. C., Ferraiolo, D. F., & Kuhn, D. R. (2014). Attribute-Based Access Control. NIST Special Publication 800-162.
5. Rose, S., Borchert, O., Mitchell, S., & Connelly, S. (2020). Zero Trust Architecture. NIST Special Publication 800-207.
6. Fernandez-Aleman, J. L., et al. (2013). Security and privacy in electronic health records: A systematic literature review. *Journal of Biomedical Informatics*, 46(3), 541–562.
7. Pearson, S. (2013). *Privacy, Security and Trust in Cloud Computing*. Springer.
8. Yu, S., Wang, C., Ren, K., & Lou, W. (2010). Achieving secure, scalable, and fine-grained data access control in cloud computing. *IEEE INFOCOM*.
9. Sahai, A., & Waters, B. (2005). Fuzzy identity-based encryption. In *Advances in Cryptology – EUROCRYPT*, Springer.
10. Lewko, A., Okamoto, T., Sahai, A., Takashima, K., & Waters, B. (2010). Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption. In *Advances in Cryptology – EUROCRYPT*.
11. KaipingXue "RAAC: Robust and Auditable Access Control with Multiple Attribute Authorities for Public Cloud Storage", *IEEE2016*.
12. Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE Transactions on Parallel & Distributed Systems*, vol. 27, no. 9, pp. 2546–2559, 2016.

13. Z. Fu, X. Sun, S. Ji, and G. Xie, "Towards efficient content-aware search over encrypted outsourced data in cloud," in in Proceedings of 2016 IEEE Conference on Computer Communications (INFOCOM 2016). IEEE, 2016, pp. 1–9.
14. K. Xue and P. Hong, "A dynamic secure group sharing framework in public cloud computing," IEEE Transactions on Cloud Computing, vol. 2, no. 4, pp. 459–470, 2014.
15. Y. Wu, Z. Wei, and H. Deng, "Attributebased access to scalable media in cloudassisted content sharing," IEEE Transactions on Multimedia, vol. 15, no. 4, pp. 778–788, 2013.
16. J. Hur, "Improving security and efficiency in attributebased data sharing," IEEE Transactions on Knowledge and Data Engineering, vol. 25, no. 10, pp. 2271– 2282, 2013.
17. J. Hur and D. K. Noh, "Attribute-based access control with efficient revocation in data outsourcing systems," IEEE Transactions on Parallel and Distributed Systems, vol. 22, no. 7, pp. 1214–1221, 2011.
18. Korra, S., Babu, A.V. and Raju, S.V., 2014, November. The adaptive approach to software reuse. In 2014 International Conference on Contemporary Computing and Informatics (IC3I) (pp. 67-70). IEEE.
19. Korra, S., Mamidi, R., Soora, N.R., Kumar, K.V. and Kumar, N.C.S., 2022. Intracranial hemorrhage subtype classification using learned fully connected separable convolutional network. Concurrency and Computation: Practice and Experience, 34(24), p.e7218.
20. Korra, S., Vasumathi, D. and Vinayababu, A., 2018, June. An approach for cognitive software reuse framework. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS) (pp. 1-6). IEEE.