International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 75-85
ISSN: 3071-3358

Secure Privacy-Preserving Search And Access Control For Encrypted Cloud Data

Mohammed Talib[1], Dr. Md Zainlabuddin[2]
[1]PG Scholar, Department Of Computer Science & Engineering ISL Engineering College, Hyderabad, India.
[2]Associate Professor; Department Of Computer Science & Engineering ISL Engineering College, Hyderabad, India
Corresponding Author Email: talib.razaa@gmail.com

Abstract
Cloud-assisted Internet of Things (IoT) applications require secure mechanisms for storing, sharing, and retrieving encrypted data. Key-aggregate searchable encryption is an effective approach that enables a cloud server to perform keyword searches over files encrypted under different public keys while allowing authorized users to use a single, constant-size aggregate secret key. However, many existing schemes remain vulnerable to keyword-guessing attacks, which can expose sensitive information contained in keyword ciphertexts and search trapdoors.
This study proposes a secure and efficient key-aggregate searchable encryption scheme for cloud-assisted IoT environments. The proposed approach enables data owners to securely share selected encrypted documents with multiple authorized users while allowing users to delegate keyword-search operations to the cloud server without revealing the privacy of the underlying keywords or trapdoors. The security of the scheme is formally defined under the standard model and demonstrated against indistinguishable selective-file chosen-keyword attacks. A rigorous security proof is provided to validate the confidentiality and privacy guarantees of the proposed mechanism. Furthermore, theoretical analysis and experimental simulations demonstrate the efficiency, flexibility, and practical applicability of the scheme for secure data retrieval in cloud-assisted IoT systems.

Keywords: *Cloud Computing, Internet of Things (IoT), Key-Aggregate Encryption, Searchable Encryption, Keyword Search, Data Privacy, Cloud Security, Keyword Guessing Attacks, Trapdoor Privacy, Secure Data Sharing.*

1. Introduction

The rapid growth of cloud computing and Internet of Things (IoT) applications has created a practical need for storing and accessing large volumes of sensitive information outside the local computing environment. Cloud platforms provide scalable storage and processing resources, but transferring confidential data to third-party infrastructure introduces concerns related to unauthorized access, information leakage, and control over stored content. Armbrust et al. highlighted the scalability and flexibility offered by cloud computing while also emphasizing the challenges associated with relying on remote infrastructure for data services [15]. These concerns become more significant in IoT environments, where devices continuously generate data and often operate with limited computational and storage capabilities. Encryption provides an important foundation for protecting cloud-stored information. However, conventional encryption makes direct searching difficult because the cloud server cannot normally interpret the encrypted contents. Searchable encryption addresses this limitation by allowing specific operations to be performed over protected data without requiring the underlying files to be decrypted. Song, Wagner, and Perrig established an early approach to searching encrypted information, demonstrating that useful search operations could be performed while maintaining confidentiality of stored data [1].

Received: *11-06-2026*
Revised: *23-07-2026*
Accepted: *07-08-2026*
Published: *19-08-2026*
Citation: *" Secure Privacy-Preserving Search And Access Control For Encrypted Cloud Data", ijaicn, vol. 2, no. 3, pp. 75–85, August. 2026,*

Public-key searchable encryption subsequently extended this concept by enabling keyword-based retrieval within public-key encryption settings [7].

The development of searchable encryption has led to more flexible schemes capable of supporting dynamic data, fuzzy queries, multi-user environments, and selective data sharing. Curtmola et al. formalized searchable symmetric encryption and presented efficient constructions for secure keyword retrieval [21]. Kamara, Papamanthou, and Roeder further addressed the dynamic nature of cloud data by developing dynamic searchable symmetric encryption, allowing encrypted collections to be updated while supporting search operations [16]. Fuzzy keyword search was also investigated by Li et al., enabling users to retrieve encrypted documents even when search terms do not exactly match stored keywords [24].

For cloud-assisted IoT systems, access control presents an additional challenge because different users may require access to different subsets of encrypted files. Key-aggregate searchable encryption provides an attractive solution by allowing a data owner to generate a compact aggregate key capable of representing access rights over multiple encrypted data objects. Cui, Liu, and Wang investigated key-aggregate searchable encryption for group data sharing through cloud storage, demonstrating how access to multiple encrypted files can be managed using aggregated cryptographic information [4]. Multi-key searchable encryption has also been explored for Industrial IoT environments, where data may be encrypted under different keys and accessed through keyword-based mechanisms [13].

Despite these developments, privacy-preserving search remains vulnerable to practical security concerns. Keyword information can become a source of leakage when search tokens or ciphertext structures reveal patterns that can be exploited by an adversary. Efficient encrypted keyword-search mechanisms for multi-user environments have therefore received considerable attention [22]. Secure data sharing in heterogeneous and edge-assisted environments additionally requires efficient communication and retrieval mechanisms because IoT devices may have limited resources [10]. Offloading and cloud-assisted processing can improve system efficiency, but security mechanisms must be integrated without creating excessive computational or communication overhead [19].

Blockchain-based access control provides another complementary direction for distributed trust management. Misbah Kousar, Sanjay Kumar, and Mohammed Abdul Bari proposed a decentralized authentication and off-chain data-management protocol for VANETs using blockchain, demonstrating the relevance of decentralized security mechanisms for distributed environments [3]. Similarly, Nishat, Mohammed Abdul Bari, and Guddi Singh investigated mechanisms for mitigating misbehaving nodes in mobile ad hoc networks, emphasizing the importance of secure communication and trustworthy participation in distributed networks [12].

Motivated by these requirements, this study proposes a **Secure Privacy-Preserving Search and Access Control framework for encrypted cloud data**. The proposed approach combines key-aggregate searchable encryption, controlled data sharing, privacy-preserving keyword search, and access-control mechanisms to allow authorized users to retrieve relevant encrypted documents without exposing sensitive keywords or trapdoors. The framework is designed particularly for cloud-assisted IoT environments, where security, scalability, search flexibility, and computational efficiency must be considered simultaneously. The proposed scheme is evaluated through security analysis, theoretical examination, and experimental simulation to determine its ability to provide protected search and controlled access while maintaining practical performance.

2. Literature Review

2.1 Cloud Computing and Secure Data Management

Armbrust et al. examined the fundamental characteristics of cloud computing and described how remotely hosted computing resources can provide scalable services to users. Their work established the broader technological foundation for cloud-based storage and computation, which has subsequently become important for data-intensive IoT applications. However, the movement of sensitive information to shared cloud infrastructure also creates a requirement for stronger mechanisms for confidentiality and access management [15].

Sannidhanam investigated scalable web-service architectures for healthcare data processing, highlighting the importance of scalable infrastructure when handling data-intensive applications. Such architectures demonstrate the value of cloud-based processing for large information workloads while also reinforcing the need to incorporate appropriate security and privacy mechanisms when sensitive information is processed remotely [2].

2.2 Search Over Encrypted Data

Song, Wagner, and Perrig introduced practical techniques for searching encrypted data without requiring the server to receive plaintext information.

Their work established an important foundation for privacy-preserving search by demonstrating how encrypted content can support useful retrieval operations while limiting direct exposure of the underlying data [1].

Boneh, Di Crescenzo, Ostrovsky, and Persiano developed public-key encryption with keyword search, extending searchable encryption into a public-key setting. Their approach demonstrated that encrypted data could be queried through keyword-related cryptographic mechanisms without directly revealing the protected content, providing a foundation for later public-key searchable-encryption schemes [7].

Curtmola, Garay, Kamara, and Ostrovsky formalized searchable symmetric encryption and developed efficient constructions for encrypted keyword retrieval. Their work provided stronger definitions of search security and helped establish a framework for analyzing leakage and adversarial capabilities in searchable encrypted databases [21].

Li et al. proposed fuzzy keyword search over encrypted cloud data to address situations in which an exact keyword match may not be available. Their approach improves retrieval flexibility by permitting searches involving similar or approximate terms, which can be useful in practical cloud environments where users may enter incomplete or slightly different keywords [24].

2.3 Dynamic and Multi-User Searchable Encryption

Kamara, Papamanthou, and Roeder investigated dynamic searchable symmetric encryption to address the requirement for updating encrypted datasets. Their work enabled encrypted collections to support operations associated with dynamic data while preserving the ability to perform keyword searches, making the approach more suitable for cloud environments where files are frequently added, modified, or removed [16].

Kiayias, Oksuz, and Russell studied encrypted keyword search in multi-user data-sharing environments. Their work focused on supporting secure keyword retrieval when encrypted information is shared among multiple users, providing an important foundation for systems in which access permissions differ between users [22].

Zhou et al. developed a file-centric multi-key aggregate keyword-searchable encryption mechanism for Industrial IoT. Their work addressed situations in which multiple encrypted files and cryptographic keys must be managed simultaneously, demonstrating the relevance of aggregate keyword-search techniques for IoT-oriented cloud environments [13].

2.4 Key-Aggregate Encryption and Access Control

Cui, Liu, and Wang proposed key-aggregate searchable encryption for group data sharing through cloud storage. Their approach allows access rights associated with multiple encrypted files to be represented through an aggregate cryptographic key, reducing the burden of managing separate keys for every individual data object. This concept is closely related to the present study because it provides a practical basis for combining selective access control with searchable encryption [4].

Miao et al. investigated lightweight fine-grained search over encrypted data in fog computing. Their work addressed the need for efficient and granular retrieval mechanisms in distributed computing environments, which is particularly relevant to IoT systems where data may be processed closer to the edge rather than exclusively within centralized cloud infrastructure [9].

Sun et al. examined mobile access and flexible search over encrypted cloud data in heterogeneous systems. Their research emphasizes the importance of supporting secure retrieval across different types of devices and network environments. Such flexibility is particularly significant in cloud-assisted IoT applications, where devices can have widely varying computational capabilities and connectivity conditions [10].

2.5 Secure Distributed and IoT Environments

Ale et al. investigated proactive caching in mobile edge computing using bidirectional deep recurrent neural networks. Their work demonstrates how edge intelligence can improve data availability and responsiveness in distributed environments. Although their primary focus is caching rather than encrypted search, the study is relevant because secure cloud-IoT frameworks must operate efficiently across cloud and edge resources [5].

Jia et al. investigated secure data transmission between IoT and cloud environments using secure truncating OFDM. Their work highlights the importance of protecting information during communication between resource-constrained IoT environments and cloud infrastructure. Secure transmission complements encrypted storage and privacy-preserving retrieval by protecting data across multiple stages of the cloud-IoT workflow [19].

Misbah Kousar, Sanjay Kumar, and Mohammed Abdul Bari proposed a blockchain-based decentralized authentication and off-chain data-management protocol for VANETs. Their research demonstrates how decentralized mechanisms can strengthen authentication and data-management processes in distributed networks. The concept of decentralized trust is relevant to privacy-preserving

cloud systems in which users, data owners, and service providers may not fully trust a single centralized authority [3].

Afsha Nishat, Mohammed Abdul Bari, and Guddi Singh examined a reactive routing mechanism for mitigating misbehaving nodes in mobile ad hoc networks. Their study emphasizes the importance of identifying and controlling untrusted participants in distributed communication environments. This perspective is relevant to secure cloud-assisted IoT systems, where unauthorized or malicious entities can threaten data confidentiality and system reliability [12].

Overall, the reviewed literature demonstrates substantial progress in searchable encryption, key aggregation, cloud security, and distributed IoT protection. However, challenges remain in simultaneously achieving fine-grained access control, privacy-preserving keyword search, trapdoor confidentiality, resistance to keyword-guessing attacks, multi-user sharing, and practical computational efficiency**.** The proposed framework addresses these requirements by integrating key-aggregate searchable encryption with privacy-preserving search and controlled access mechanisms for encrypted cloud data.

3. Methodologies

The proposed system consists of five major modules. Key-Aggregate Searchable Encryption (KASE) allows authorized users to perform searches across multiple encrypted files using a compact aggregate key, thereby reducing key-management overhead. AES-256 provides strong symmetric encryption for protecting sensitive file contents. The cloud server stores encrypted files and performs secure search operations without exposing confidential information. The JSP Dashboard offers a web-based interface for file management, searching, viewing results, and accessing system services. The MySQL database stores user information, file metadata, access permissions, and other application-related records required for authentication and system management.

Requirements Engineering

The Secure Keyword Search and Key Management Scheme provides a privacy-preserving mechanism for storing, sharing, and retrieving encrypted cloud data. The system combines key-aggregate searchable encryption, secure trapdoor generation, access control, and cloud storage to enable efficient keyword searching without exposing sensitive information. It allows data owners to share multiple documents using a compact secret key while protecting keyword ciphertexts and trapdoors against keyword-guessing attacks. The framework is designed to support secure access, scalability, and efficient data management in cloud and IoT environments.

Hardware Requirements

The system requires standard computing hardware for development and deployment. The minimum configuration includes an Intel Core i3 or higher processor, 4 GB DDR4 or higher RAM, a 15.6-inch LED monitor, at least 100 GB of hard-disk storage, and standard input devices such as a keyboard and mouse. A webcam may also be used where required by additional system functions.

Software Requirements

The software environment consists of Java EE technologies, including JSP, Servlets, and Maven, for application development. MySQL 5.5 or above is used for database management, while Windows 10 or Windows 11 provides the operating environment. Apache Tomcat 9.0 is used as the web server, and Google Chrome or Mozilla Firefox supports browser-based access. Eclipse IDE or IntelliJ IDEA can be used for application development and testing.

4. Design Engineering

Design Engineering defines the structure, behavior, and interaction of the proposed secure keyword-search framework using Unified Modeling Language (UML). The design represents the relationships among data owners, data users, administrators, cloud servers, encrypted files, and key-management components. Use case diagrams describe major functions such as registration, authentication, file upload, key generation, search requests, access approval, and secure downloading. Class diagrams define the core system entities and their relationships, while sequence and activity diagrams illustrate the flow of encryption, trapdoor generation, cloud-based searching, and data retrieval. Component and deployment diagrams describe the interaction and distribution of the encryption module, KASE module, cloud storage, access-control services, and database. Together, these diagrams provide a structured blueprint for developing a secure, scalable, and privacy-preserving cloud data-search system.

System Architecture

International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 75-85
ISSN: 3071-3358

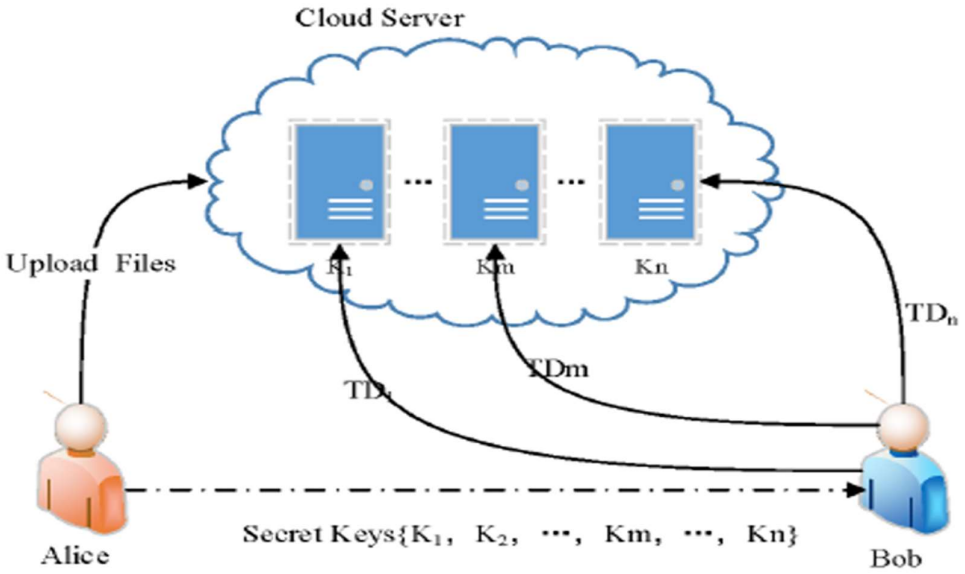

Fig 1: System Architecture

The system architecture consists of three primary entities: the Data Owner, Data User, and Administrator. Users register and authenticate before accessing the system's services. The Data Owner uploads files, applies encryption, and defines sharing permissions, while the Data User submits search and cloud-service requests to locate authorized data. A central database maintains metadata, file information, and access records, whereas the cloud server manages large-scale encrypted storage and search processing. The Administrator supervises registrations and sharing permissions. Secure key distribution between owners and authorized users ensures that encrypted data remains protected during cloud storage and retrieval.

Development Tools

The proposed application is developed using Java technologies, with Java EE selected as the primary implementation platform. Java provides object-oriented programming capabilities, platform independence, security, portability, and support for developing networked and web-based applications. Java EE extends these capabilities by providing technologies suitable for enterprise and server-side application development.

Features Of Java

The Java Framework

Java is a general-purpose, class-based, object-oriented programming language designed for portability and platform independence. Java programs are compiled into bytecode that executes on the Java Virtual Machine (JVM), allowing applications to operate across different hardware and operating systems. Its security, concurrency support, portability, and extensive libraries make Java suitable for desktop, web, enterprise, mobile, and distributed applications.

Objectives Of Java

Java enables developers to create software on one platform and execute it on different systems with minimal modification. It supports web applications, server-side processing, online services, mobile applications, and distributed systems. Its object-oriented structure promotes code reuse and modular development through inheritance, encapsulation, polymorphism, and dynamic binding, making applications easier to develop, maintain, and extend.

Java Swing Overview

Java Swing provides a collection of graphical user-interface components for developing desktop applications. It includes components such as buttons, lists, frames, and dialogs, with most components implemented as lightweight elements rendered through Java. Swing components are built on top of the Abstract Window Toolkit (AWT), while top-level containers such as JFrame and JDialog

International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 75-85
ISSN: 3071-3358

provide the connection between the application interface and the operating system.

Evolution Of Collection Framework

The Java Collection Framework provides standardized interfaces and classes for storing and managing groups of objects. The Collection interface defines common operations such as adding, removing, checking, and iterating over elements. The main collection types are Lists, Sets, and Maps. Lists maintain ordered elements and allow duplicates, Sets store unique elements, and Maps associate unique keys with corresponding values. Implementations such as ArrayList, LinkedList, HashSet, TreeSet, HashMap, LinkedHashMap, and TreeMap provide different mechanisms for efficient data storage and retrieval.

Threads

A thread represents an independent path of execution within a program. Multithreading enables multiple operations to execute concurrently, improving application responsiveness and efficiency. In Java, threads can be created by extending the Thread class or implementing the Runnable interface. The interface-based approach is commonly preferred because it provides greater flexibility when designing applications.

Advanced Java

Advanced Java includes technologies and libraries used to develop web-based, enterprise, and distributed applications. Servlets handle server-side HTTP requests and responses, while JavaServer Pages (JSP) support the creation of dynamic web content. Java Database Connectivity (JDBC) provides database connectivity and query execution, whereas JavaBeans support reusable application components. The Model-View-Controller architecture separates application logic, presentation, and control. Filters and listeners manage request processing and application events, while session-management techniques such as cookies, HttpSession, and URL rewriting maintain user information across multiple web pages.

Apache Tomcat Server

Apache Tomcat is an open-source server and servlet container used to deploy Java web applications. It supports technologies such as Servlets, JSP, and HTTP communication, providing an environment for executing server-side Java applications. Tomcat is lightweight, easy to configure, supports WAR-file deployment, and integrates with development environments such as Eclipse and IntelliJ IDEA.

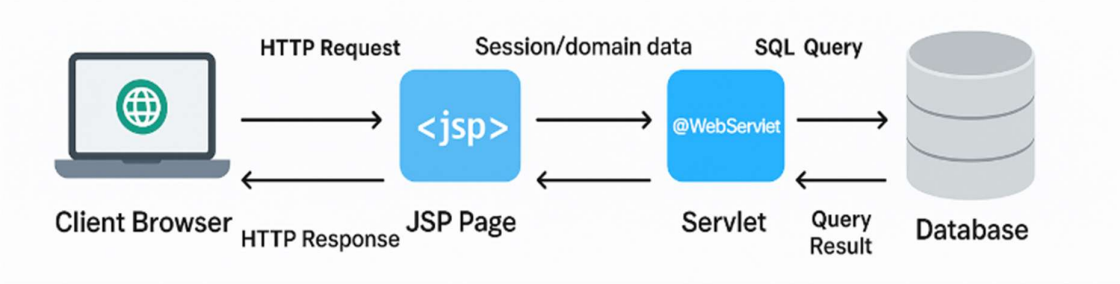

Fig 2

5. Implementation

The flowchart illustrates the implementation workflow of the Secure Privacy-Preserving Search and Access Control for Encrypted Cloud Data system. The process begins with the user opening the application and selecting either the Login or Owner Registration module.

In the Login module, the user provides a username and password. The servlet retrieves and verifies the credentials. If the credentials are valid, a session is created and the user is redirected to the cloud dashboard. If the credentials are invalid, an error message is displayed and the user is redirected to the login page.

In the Owner Registration module, the user enters registration details, including name, email, password, mobile number, gender, and address. The servlet validates the mobile number before establishing a connection with the MySQL database. Valid information is inserted into the owners table. If the email already exists, an error message is displayed; otherwise, a successful registration message is generated.

Finally, all successful and unsuccessful processing paths converge at the **End** stage. The flowchart therefore demonstrates the complete application workflow, including user authentication, registration, input validation, database interaction,

International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 75-85
ISSN: 3071-3358

session management, error handling, and successful access control.

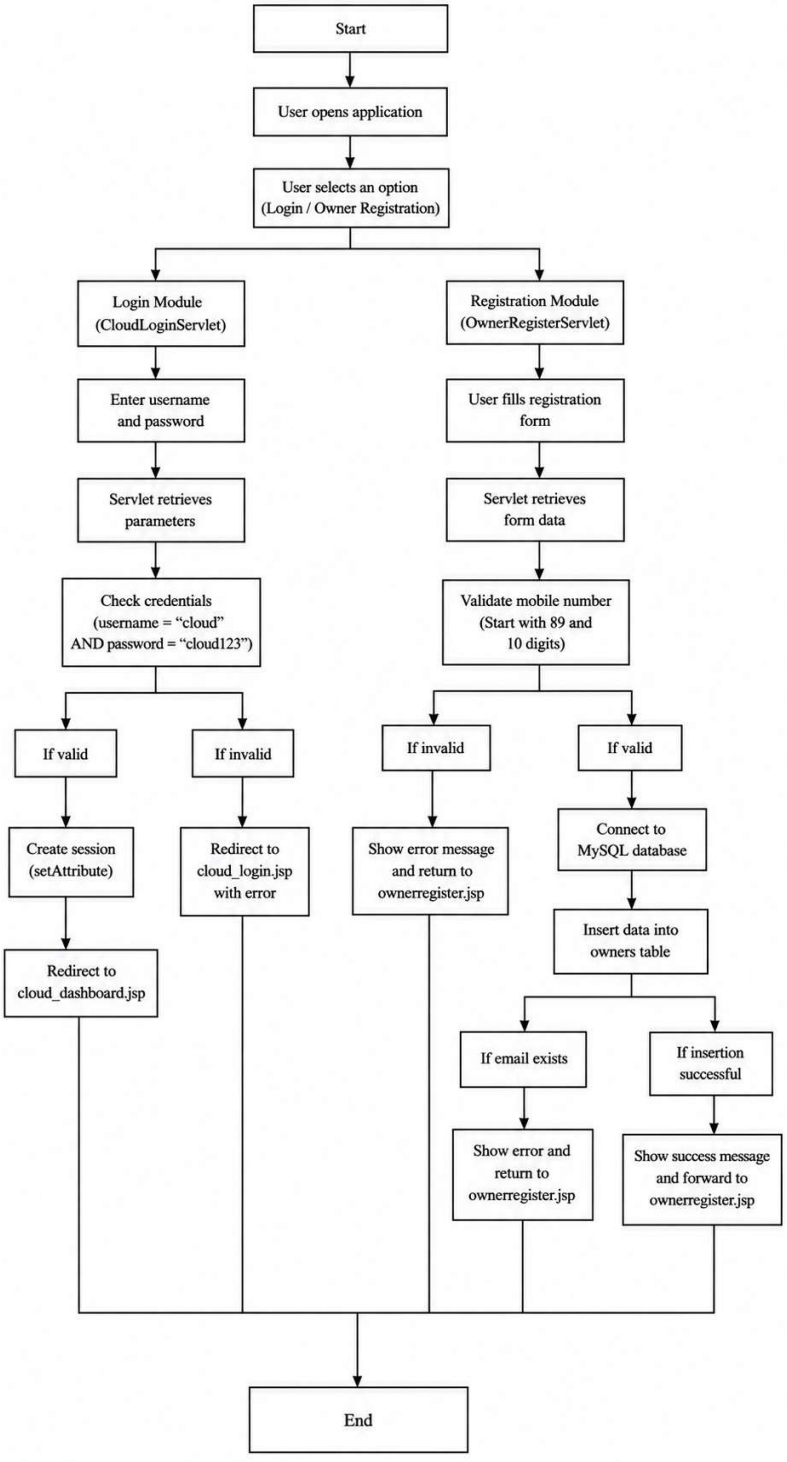

Figure 3: Implementation Process

International Journal of Artificial Intelligence and Communication Networks
Volume: 02, Issue: 03, July-September 2026, pp: 75-85
ISSN: 3071-3358

6. Results and Snapshots

This project is implements like web application using COREJAVA and the Server process is maintained using the SOCKET & SERVERSOCKET and the Design part is played by Cascading Style Sheet.

Various Snapshots

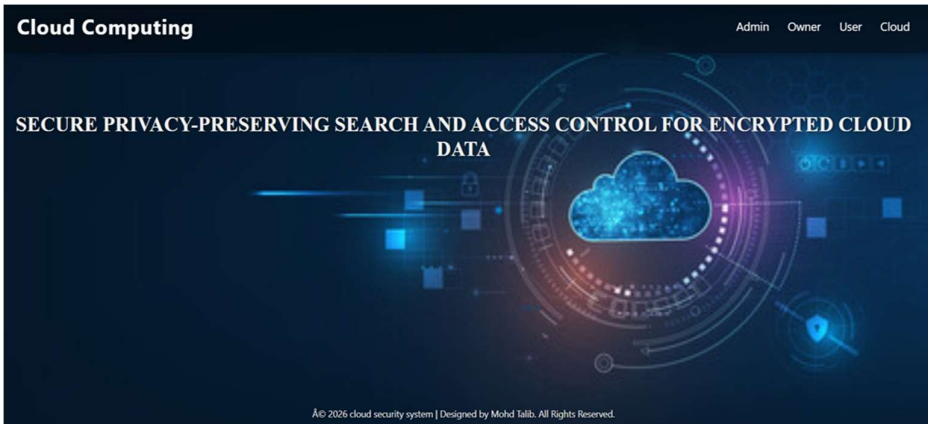

Fig 4

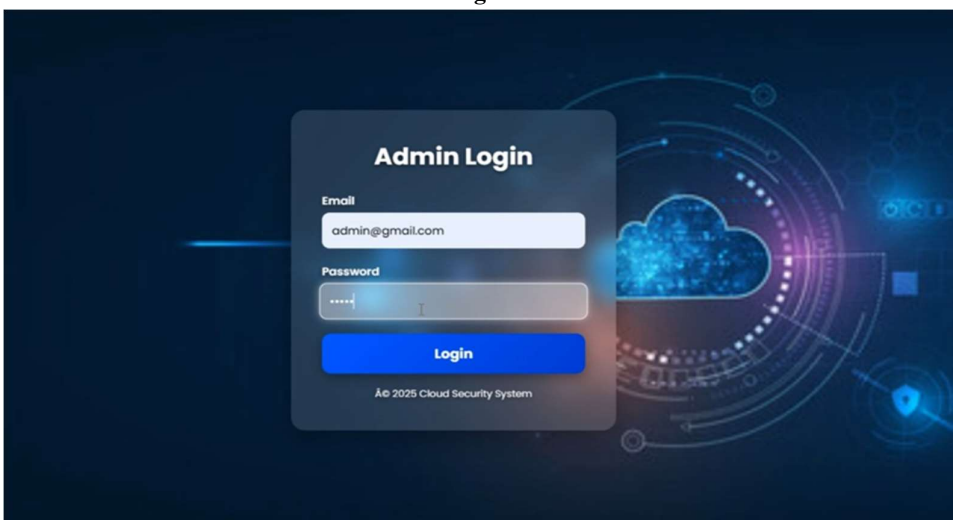

Fig 5

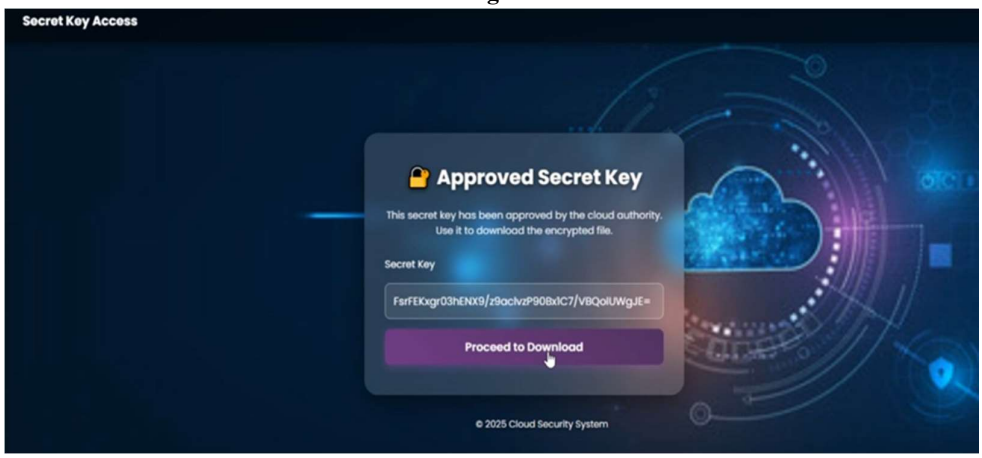

Fig 6

Software Testing

Software testing is performed to identify errors, defects, and weaknesses in an application and to verify that the developed system satisfies its specified requirements and user expectations. It evaluates individual components as well as the complete integrated system to ensure reliable and acceptable operation. Different testing methods are applied to examine specific functional and non-functional requirements.

Developing Methodologies

The testing process begins with a comprehensive test plan covering the application's general functions, special features, and operational requirements across relevant platforms. Quality-control procedures are applied to verify that the system complies with the requirements specified during development and that identified defects are corrected. The testing methodology provides a structured framework for evaluating the reliability and functionality of the application.

7. Conclusion

This study presents a secure and efficient framework for keyword searching and key management in cloud environments, with particular relevance to cloud-assisted IoT applications. The proposed system addresses the challenge of securely storing, sharing, and retrieving sensitive information when data is outsourced to an external cloud infrastructure.

The proposed Key-Aggregate Searchable Encryption (KASE) mechanism allows data owners to encrypt and selectively share multiple documents while enabling authorized users to perform keyword searches using a single compact aggregate key. This approach reduces the complexity of managing separate keys for individual files and supports controlled access to selected documents.

A major focus of the scheme is the protection of keyword information against guessing attacks. By safeguarding keyword ciphertexts and search trapdoors, the system improves search privacy and reduces the possibility of sensitive information being inferred by unauthorized entities or the cloud server.

The framework also supports delegated searching, allowing the cloud server to search encrypted documents without requiring users to download and decrypt all stored files. AES-256 provides confidentiality for sensitive data, while the cloud server manages encrypted storage and retrieval. The JSP-based interface supports user interaction, and the MySQL database manages user information, metadata, and access-related records.

References

1. D. Song, D. Wagner, and A. Perrig, “Practical Techniques for Searches on Encrypted Data,” in *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 44–55, 2000.
2. Sannidhanam, A. H. (2020). Scalable Web Services Architecture for Healthcare Data Processing. *International Journal of Recent Advances in Science and Technology, 7*(01), 1–14.
3. Mrs. Misbah Kousar , Dr. Sanjay Kumar , Dr. Mohammed Abdul Bari**,”** *Design of a Decentralized Authentication and Off-Chain Data Management Protocol for VANETs Using Blockchain***”,** Communications on Applied Nonlinear Analysis, ISSN: 1074-133X, Vol 32 No. 2(2025)
4. B. Cui, Z. Liu, and L. Wang, “Key-Aggregate Searchable Encryption for Group Data Sharing via Cloud Storage,” *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2374–2385, 2016.
5. L. Ale, N. Zhang, H. Wu, D. Chen, and T. Han, “Online Proactive Caching in Mobile Edge Computing Using Bidirectional Deep Recurrent Neural Networks,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 5520–5530, 2019.
6. Anjani Haritha Sannidhanam. (2023). Self-Healing Distributed Systems: AI-Driven Failure Prediction and Automated Recovery. *International Journal of Research & Technology, 11*(4), 168–174.

7. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public Key Encryption With Keyword Search," in *Advances in Cryptology—EUROCRYPT*, pp. 506–522, 2004.
8. Sannidhanam, A. H. (2025). Autonomous AI Agents for Cloud Infrastructure Operations. *Journal of Contemporary Science and Technology Management, 1*(01), 83–100.
9. Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, and H. Li, "Lightweight Fine-Grained Search Over Encrypted Data in Fog Computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 772–785, 2019.
10. J. Sun, H. Xiong, H. Zhang, and L. Peng, "Mobile Access and Flexible Search Over Encrypted Cloud Data in Heterogeneous Systems," *Information Sciences*, vol. 507, pp. 1–15, 2020.
11. Distributed Data Processing Frameworks for Large-Scale Healthcare Analytics. (2019). *International Journal of Advance Industrial Engineering, 7*(04), 1–10. doi:10.14741/ijaie/v.7.4.01
12. Afsha Nishat, Dr. Mohd Abdul Bari, Dr. Guddi Singh," *Mobile Ad Hoc Network Reactive Routing Protocol to Mitigate Misbehavior Node*", International Journal Of Intelligent Systems And Applications In Engineering, IJUSEA, ISSN:2147-6799, Nov 2023.
13. R. Zhou, X. Zhang, X. Du, X. Wang, G. Yang, and M. Guizani, "File-Centric Multi-Key Aggregate Keyword Searchable Encryption for Industrial Internet of Things," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3648–3658, 2018.
14. Anjani Haritha Sannidhanam. (2024). Guardrails and Safety Mechanisms for LLM-Powered Enterprise Applications. *International Journal of Engineering Science & Humanities, 14*(3), 273–285.
15. M. Armbrust *et al.*, "A View of Cloud Computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
16. S. Kamara, C. Papamanthou, and T. Roeder, "Dynamic Searchable Symmetric Encryption," in *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 965–976, 2012.
17. Sannidhanam, A. H., & Alladi, S. (2017). Cloud-Based Healthcare CRM Systems for Improved Patient Engagement. *International Journal of Technology, Management and Humanities, 3*(01), 32–44. doi:10.21590/ijtmh.3.03.4
18. Dr.Abdul Bari ,Dr. Imtiyaz khan , Dr. Rafath Samrin , Dr. Akhil Khare , " *VPC & Public Cloud Optimal Perfomance in Cloud Environment* ",Educational Administration: Theory and Practic,ISSN No : 2148-2403 Vol 30- Issue -6 June 2024.
19. M. Jia, Z. Yin, D. Li, Q. Guo, and X. Gu, "Toward Improved Offloading Efficiency of Data Transmission in the IoT-Cloud by Leveraging Secure Truncating OFDM," *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4252–4261, 2019.
20. Event Streaming Architectures for High-Volume Transaction Processing. (2022). *International Journal of Advance Industrial Engineering, 10*(01), 1–8. doi:10.14741/
21. R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions," in *Proceedings of the ACM Conference on Computer and Communications Security*, 2006.
22. A. Kiayias, O. Oksuz, and A. Russell, "Efficient Encrypted Keyword Search for Multi-User Data Sharing," in *European Symposium on Research in*

Computer Security, pp. 173–195, 2016.

23. M. Jia, D. Li, Z. Yin, Q. Guo, and X. Gu, "High Spectral Efficiency Secure Communications With Nonorthogonal Physical and Multiple Access Layers," *IEEE Internet of Things Journal*, vol. 6, no. 4, pp. 5954–5961, 2019.
24. J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy Keyword Search Over Encrypted Data in Cloud Computing," in *Proceedings of IEEE INFOCOM*, pp. 1–5, 2010.
25. Sannidhanam, A. H. (2025). Autonomous AI Agents for Cloud Infrastructure Operations. *Journal of Contemporary Science and Technology Management, 1*(01), 83–100.
26. Dr.Abdul Bari ,Dr. Imtiyaz khan , Dr. Rafath Samrin , Dr. Akhil Khare , " *VPC & Public Cloud Optimal Perfomance in Cloud Environment* ",Educational Administration: Theory and Practic,ISSN No : 2148-2403 Vol 30- Issue -6 June 2024.

About Author:

Mohammed Talib received his Bachelor's degree in Computer Science and Engineering from T.K.R. College of Engineering & Technology, Hyderabad, India, in 2023. He is currently pursuing his Master's degree in Computer Science at ISL Engineering College, Hyderabad, India. His areas of interest include software engineering, data science, machine learning, cloud computing, backend development, and semantic web technologies. His technical interests also include Python, AWS cloud infrastructure, and software testing.