Shodhshauryam, International Scientific Refereed Research Journal
Available online at : www.shisrrj.com
OPEN ACCESS
© 2024 SHISRRJ | Volume 7 | Issue 2
ISSN : 2581-6306
doi : https://doi.org/10.32628/SHISRRJ

Privacy Preserving Multi Keyword Searchable Encryption for Distributed Systems

T. Raja Sekhar[1] , Vathaluru Narasa Reddy Reshma[2]

[1]Assistant Professor, Department of MCA, Annamacharya Institute of Technology & Sciences, Tirupati, Andhra Pradesh, India
[2]Post Graduate, Department of MCA, Annamacharya Institute of Technology & Sciences, Tirupati, Andhra Pradesh, India

Article Info

Article History
Received : 02 April 2024
Published : 13 April 2024

Publication Issue :
March-April-2024
Volume 7, Issue 2

Page Number : 486-492

ABSTRACT

With the increasing prevalence of distributed systems and the growing concern for privacy in data storage and retrieval, there is a pressing need for efficient and secure solutions. This paper introduces a (PPMKSE) scheme tailored for distributed systems, implemented using the Python programming language. With the rising prevalence of cloud storage across diverse applications, ensuring privacy during searching and retrieve data in a distributed environment presents a significant research challenge. Encryption schemes available for searching are still fall short in terms of working, privacy and security aspects. Specifically, supporting multi-user and keyword search, providing search and access patterns, and thwarting keyword guess attacks (KGA) are among the daunting tasks. We introduce a valance encryption & searchable scheme that effectively describes the aforementioned challenges simultaneously in this application. Rendering it suitable for adoption in distributed systems. Our proposed solution not only provides multiple keyword searches and multiple writers or multiple readers setting but also ensures privacy of both the data and the search pattern. In our scheme, we have devised a subset decision mechanism as the cornerstone technique, which extends its utility beyond keyword search applications. Lastly, we validate the security, assess the computational efficiency, and calculate the communication efficiency to demonstrate its practicality.

Keywords : Multi-Keyword Search, Search Pattern, Access Pattern, Searchable Encryption, Multi-User Access.

I. INTRODUCTION

In an era dominated by vast amounts of distributed data, ensuring the privacy of sensitive information has become a critical concern. As organizations and individuals continue to rely on distributed systems for data storage and retrieval, the need for robust security measures has intensified. This paper introduces a ground breaking approach to address these concerns through the development of a PPMKSE system specifically designed for systems which are distributed.

The objective of developing of the system for distributed systems is to search efficiently on encrypted data while preserving sensitive information with privacy. Specifically, the key objectives include: Privacy Preservation, Multi-Keyword Search ability, Efficiency, Scalability, Security Guarantees, Flexibility and Usability, Interoperability, Compliance, Robustness, Research Contribution

II. METHODS AND MATERIAL

The methodology utilized for enhancing PPMKSE for Distributed Systems adopts a systematic approach to ensure the successful implementation and deployment of the application. It comprises several key stages, including requirements gathering, design, development, testing, and deployment.

Problem Identification and Formulation: The methodology begins with clearly identifying the security and efficiency of keyword searches on distributed data while preserving privacy. This involves formulating specific objectives and requirements for the system.

Algorithm Design: Novel algorithms are developed or existing ones are adapted to list the challenges of PPMKSE in distributed systems. This involves designing encryption schemes, indexing structures, and search algorithms optimized for distributed environments.

Prototype Implementation: The designed algorithms are implemented in a prototype system using appropriate programming languages and frameworks. This involves coding the algorithms and integrating them into a distributed system architecture.

Testing and Evaluation: The prototype system is rigorously tested under various scenarios to evaluate its performance, security, and scalability. Benchmark datasets and performance metrics are used to assess the effectiveness of the system.

Security Analysis: A comprehensive security analysis is performed to identify potential vulnerabilities and ensure that the system meets desired security properties, such as confidentiality, integrity, and resistance to various attacks.

Comparison with Existing Methods: The proposed methodology is compared with existing approaches in terms of performance, security, and other relevant criteria. This helps to assess its novelty and effectiveness.

Validation & Verification: The methodology is validated through peer review, experimentation, and real-world use cases to ensure its robustness and applicability in practical scenarios.

III. SOFTWARE REQUIREMENT SPECIFICATION

The software program requirements Specification (SRS) for the enhancing PPMKSE for Distributed Systems outlines practical and non-useful necessities important for the a success development and deployment of the application.

Practical Necessities:

Consumer Authentication: The app need to allow customers to sign up, login, and logout securely. Enhancing user satisfaction by providing intuitive interfaces, clear feedback, and customizable search options

Admin Panel: An admin panel should be to be had for dealing with consumer bills, equipment listings, condo transactions, and resolving disputes.

Security: Ensuring that the encryption scheme provides robust security guarantees to protect sensitive data from unauthorized access, even in a distributed environment.

Efficiency: Implementing encryption and search algorithms that are efficient in communication and calculation, allowing for fast and scalable keyword searches over distributed data.

Privacy Preservation: Guaranteeing that the searchable encryption scheme preserves the privacy of users' queries and data contents, preventing any leakage of sensitive information to unauthorized parties.

Cryptographic primitives: Select appropriate cryptographic primitives such as symmetric or asymmetric encryption, hash functions, and homomorphic encryption based on your requirements.

Searchable encryption technique: Choose a suitable searchable encryption technique (e.g., deterministic, probabilistic, order-preserving encryption) that supports multi-keyword search while preserving privacy.

Distributed architecture: Design the scheme to operate efficiently in a distributed systems, considering factors of communication issues, distribution of data, and fault tolerance.

Non-functional requirements:

Overall performance: The app have to be responsive and performance, imparting a easy person revel in even beneath heavy load situations.

Security: Ensuring data confidentiality, integrity, and availability, as well as protection against unauthorized access, tampering, and data breaches.

Performance: System should be able to provide fast results for search queries and data retrieval operations, even under heavy loads and concurrent access.

Interoperability: Ensuring compatibility and seamless integration with existing distributed systems, databases, and applications.

Reliability: Providing high availability and fault tolerance to ensure uninterrupted access to data and services, even in the event of failures or disruptions.

Maintainability: Facilitating easy maintenance, updates, and enhancements to the system, including documentation, code refactoring, and debugging.

Compliance: Following to relevant requirements, standards of industry, and best practices related to data security and privacy, such as GDPR, HIPAA, and ISO/IEC 27001.

Auditability: Supporting logging, monitoring, and auditing capabilities to track user activities, system events, and data access for compliance and accountability purposes.

Cost-effectiveness: Minimizing infrastructure costs, licensing fees, and operational expenses while maximizing the value delivered by the system.

IV. PROPOSED SYSTEM

The proposed device Our solution involves the development of a novel searchable encryption scheme using Python.

The system will not only support multi-keyword searches within a multi-user framework but also guarantee the privacy of both data and search patterns.

To counter keyword guessing attacks, we adopt a multi-server architecture, promoting search speed, load & minimizing key leakage risks.

The primary objective of the proposed system is to develop a comprehensive solution that ensures the privacy preservation of sensitive data in distributed systems while allowing efficient multi-keyword searches. The system aims to strike a balance between security and usability, enabling users to securely search encrypted data distributed across multiple nodes.

Readability and Accessibility: The structure of the proposed system will be designing out using the Python programming language. Python's readability and versatility make the system accessible to a wider audience, facilitating adoption, experimentation, and collaboration.

Integration with Existing Python Libraries: Existing Python libraries for cryptography, distributed computing, and data structures will be leveraged to streamline development and ensure compatibility. Libraries such as PySyft for privacy-preserving distributed computing and PyCryptodome for cryptographic operations will enhance the efficiency of the implementation.

Probabilistic Data Structures: Efficient and secure indexing will be achieved through the use of probabilistic data structures like Bloom filters. This allows for efficient keyword searches without exposing the actual content of the data, preserving privacy during search operations.

Dynamic Indexing: The system will be designed to handle dynamic data changes, such as updates and deletions, in a secure manner while maintaining the privacy of the index. Protocols will be implemented to efficiently update and delete entries without compromising data privacy.

Dynamic Indexing Handling: The system is designed to handle dynamic data changes, including updates and deletions, in a secure manner while preserving the privacy of the index. Protocols are implemented to efficiently update and delete entries without compromising data privacy. This ensures that the system remains agile and responsive to evolving data requirements.

Readability and Accessibility with Python: The implementation of the proposed system using the Python programming language enhances readability and accessibility. Python's clear syntax and versatility make the system approachable to a wide audience, promoting adoption, experimentation, and collaboration among developers, researchers, and users.

Integration with Existing Python Libraries:

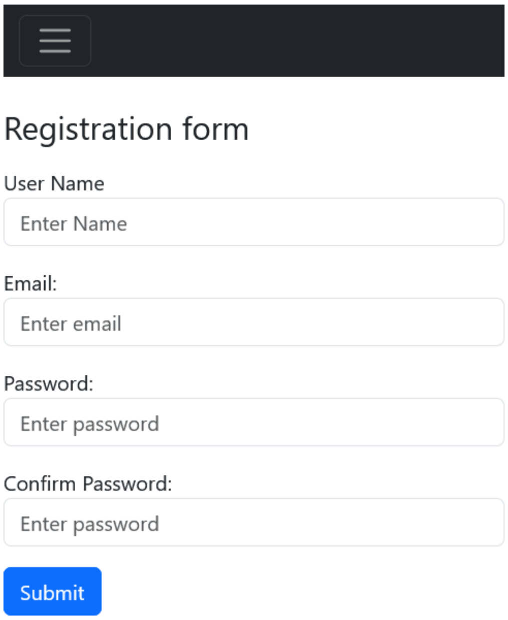
Leveraging existing Python libraries for cryptography, distributed computing, and data structures streamlines development and ensures compatibility. Integrating tools like PySyft for privacy-preserving distributed computing and PyCryptodome for cryptographic operations enhances the efficiency and effectiveness of the system.

In proposing a PPMKSE system for distributed systems, it's essential to outline the key features, components, and methodologies that differentiate it from existing systems.

The proposed PPMKSE system for distributed systems integrates advanced cryptographic techniques, distributed architecture, secure indexing, Python implementation, ledger integration, dynamic data handling, user authentication, and thorough performance evaluation. The emphasis on documentation and community engagement ensures transparency, collaboration, and ongoing development.

The system is designed for creating a user-friendly experience in a Privacy-Preserving Multi-Keyword Searchable Encryption (PPMKSE) system conceptual outline for the UI design using Python.

User interface pages are designed for interaction point for users and set the tone for the application with a design and hero section for navigation with home, search, upload and manage account. An intuitive authentication process ensures secure access to the system with a Login Form: where the users are registered using Registration Form to upload the data.



Registration form

User Name

Enter Name

Email:

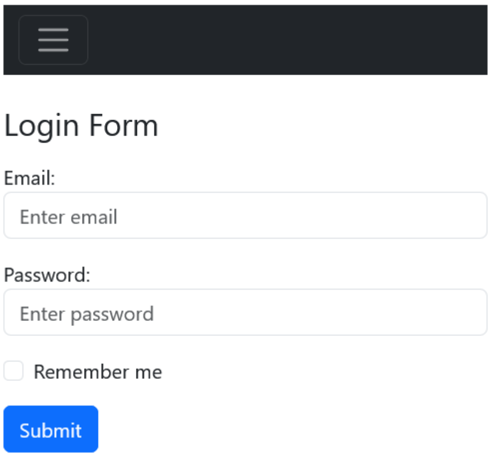
Enter email

Password:

Enter password

Confirm Password:

Enter password



Login Form

Email:

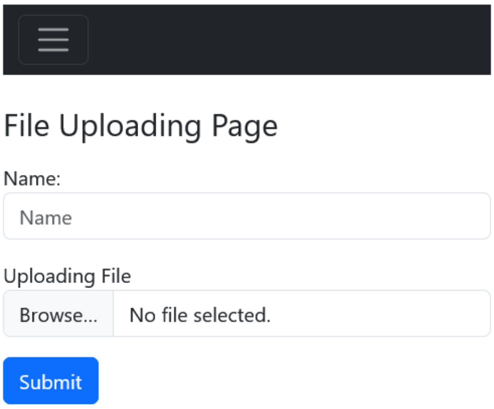
Enter email

Password:

Enter password

□ Remember me

Users can securely upload data to the system by File Upload Section: file input for uploading data files. Display a modal confirming successful data upload. Option to upload more data or navigate to other sections.



File Uploading Page

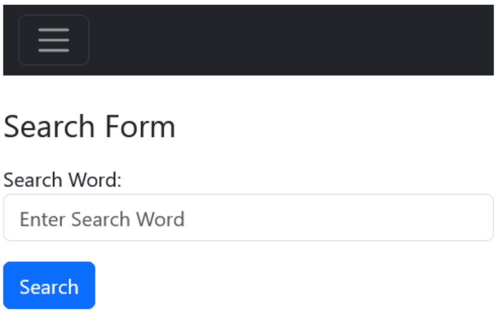
Name:

Name

Uploading File

Browse... No file selected.

Users Search with a friendly environment interface for initiating and viewing search results. By the Search Bar: Input field for multi-keyword search queries. Press the "Search" button to start the search.. Display encrypted data results, Thumbnail or metadata preview., Pagination for multiple results.



Search Form

Search Word: Enter Search Word

Users have Data Retrieval option with an interface for users to retrieve and decrypt specific data. Show comprehensive details regarding the selected data Options to initiate data retrieval.

User enables the notification as Informative messages for user feedback. Success Messages: Display success messages for actions like data upload, search, or decryption. Show error messages for unsuccessful actions with guidance on resolution.

V. ACKNOWLEDGEMENT

Acknowledging the efforts and contributions of individuals and organizations involved in the development of the Systems app is crucial to recognize their support and assistance throughout the project. Here's a sample paragraph for the acknowledgment section:

"I extend my heartfelt thanks to all individuals who have helped me to complete the project successful". First, I am very thankful to [Supervisor/Project Manager's], whose guidance, encouragement, and valuable to complete the project. I am very thankful to the total team involved in the development process, including developers, designers, testers, and support staff, for their dedication, hard work, and collaboration. Special thanks are due to [Organization/Institution's Name] for providing the necessary resources, facilities, and support throughout the project duration. I would like to extend grateful thanks to all the stakeholders, including, equipment owners, and users, whose feedback and input were invaluable in shaping the features and functionalities of the app."

VI. CONCLUSION

In conclusion, the design of a PPMKSE system for distributed systems using Python is a multifaceted endeavor that requires careful consideration of security, functionality, and efficiency. This design aims to address the unique challenges associated with preserving user privacy while enabling effective and secure multi-keyword searches across a distributed network.

Key Considerations and Achievements are Security-Centric Approach, User-Centric Design, Distributed System Architecture, Database Design, Documentation and Testing

III. REFERENCES

[1]. Curtmola, R., Garay, J., Kamara, S., & Ostrovsky, R. (2006). Searchable symmetric encryption: improved definitions and efficient constructions. Proceedings of the 13th ACM conference on Computer and communications security. CCS '06.

[2]. Cao, N., Wang, C., Li, M., Ren, K., & Lou, W. (2014). Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Transactions on Parallel and Distributed Systems, 25(1), 222-233.

[3]. Kamara, S., Papamanthou, C., & Roeder, T. (2012). Dyna2mic searchable symmetric encryption. Proceedings of the 2012 ACM conference on Computer and communications security. CCS '12.

[4]. Zhang, Y., Deng, R. H., Liu, X., & Zheng, D. (2015). Verifiable privacy-preserving multi-keyword text search in the cloud supporting similarity-based ranking. ACM Transactions on Privacy and Security (TOPS), 18(3), 1-30.

[5]. Fu, Z., Wu, X., Guan, C., Sun, X., & Ren, K. (2016). Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement. IEEE Transactions on Information Forensics and Security, 11(12), 2706-2716.

[6]. Wang, B., Song, L., Li, Q., Li, H., & Xiang, Y. (2019). A blockchain-based privacy-preserving multi-keyword search scheme. Future Generation Computer Systems, 94, 541-550.