IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Dynamic Searchable Encryption With Leakage Resilience And Verification

[1]Vepuri Roshini, [2]K Pavan Kumar, [3]Sakala Satya Sai Balaji Kumar, [4]Kamireddy Sudha Kiran, [5]Karanam Ganesh

[1]Cyber Security, [2]Cyber Security, [3]Cyber Security, [4]Cyber Security, [5]Cyber Security
Raghu Engineering College, India

Abstract: Protecting sensitive data from potential dangers is crucial in the current digital world. Since cloud computing has become widely used and data storage has been outsourced, it has become more difficult to safeguard information, especially from threats like memory leaks. This effort presents a novel cryptographic architecture called "Memory Leakage-Resilient Dynamic and Verifiable Multi-keyword Ranked Search on Encrypted" (MLRDV-MKRSE) in answer to these difficulties. The primary aim of this project is to design and implement a safe and effective solution that tackles vulnerabilities resulting from memory leaks. Additionally, this system need to provide multi-keyword ranked searches on encrypted data that may be verified and support dynamic data activities. In order to maintain data confidentiality, integrity, and availability even in the event of memory leakage occurrences, sophisticated cryptographic techniques will be utilized. Strong encryption algorithms to prevent unwanted access, flexible data management systems to meet changing data requirements, and verifiable search features that guarantee quick access to relevant data without sacrificing privacy are just a few of the essential components of the suggested plan. To assess the performance, security, and practicality of the suggested solution, extensive testing and assessment will be carried out. The goal of this research is to greatly improve safe data management methods by creating the Memory Leakage-Resilient Dynamic and Verifiable Multi-keyword Ranked Search on Encrypted scheme. In the end, it aims to provide a workable solution for safeguarding private data in cloud settings and other data outsourcing situations.

Index Terms **– Cryptographic Architecture, Encrypted, Dynamic data,**

I. INTRODUCTION

Given the current state of affairs in the digital transformation era, the efficient management and protection of sensitive data have emerged as an issue of fundamental importance. In light of the broad adoption of cloud computing and data outsourcing, the implementation of measures to protect the confidentiality and privacy of information has become increasingly complex. Memory leaking is one of the most significant dangers to data security. Memory leakage occurs when sensitive information that is stored in volatile memory is vulnerable to being accessed by unauthorized parties. The confidentiality and integrity of the data are put in jeopardy as a result of this leakage, which can be caused by a variety of circumstances, such as software vulnerabilities, hardware problems, or hostile intrusions. In order to confront these issues head-on, this project suggests the development of an all-encompassing solution that will be referred to as "Memory Leakage-Resilient Dynamic and Verifiable Multi-keyword Ranked Search on Encrypted" (MLRDV-MKRSE). Through the facilitation of dynamic data operations and the facilitation of verified multi-keyword ranked searches on encrypted data, its major purpose is to alleviate the vulnerabilities associated with memory leaking. Even if there is a possibility of memory leakage, the MLRDV-MKRSE scheme that is being envisioned works toward the goal of establishing a robust framework that guarantees the availability, integrity, and secrecy of data. In order to create a data management system that is both secure and efficient, it is necessary to incorporate sophisticated cryptography methods, dynamic data management protocols, and search features that are efficient.

This project is significant because it has the ability to improve data security in cloud computing environments and other contexts where data outsourcing is popular. This is the reason why this endeavor is essential. The MLRDV-MKRSE scheme is designed to answer the urgent necessity for robust data security methods in today's networked landscape. It does this by eliminating the risks that are posed by memory leakage and by enabling data operations that are both secure and efficient. As we move on with the project, we will be delving into the technical design, implementation, and evaluation of the MLRDV-MKRSE system. The fundamental cryptographic principles, dynamic data management strategies, and verifiable search algorithms that are utilized in the proposed solution will be investigated in depth. By conducting exhaustive experiments and conducting in-depth analyses, the project intends to establish the MLRDV-MKRSE scheme's effectiveness, efficiency, and practical viability in settings that are representative of the real world. This project's primary objective is to enhance secure data management techniques. Its goal is to provide a comprehensive solution that will protect sensitive information from memory leakage vulnerabilities and guarantee the confidentiality and integrity of data in computer environments that are both dynamic and dispersed.

II. LITERATURE REVIEW

In the presentation titled "Practical Threshold Signatures" that Victor Shoup gave at EUROCRYPT 2000, he covers the practical implementation of threshold signatures, which requires a group to generate signatures collectively for the purpose of creating a higher level of security. It presents strategies for overcoming obstacles, with a particular focus on distributed key generation (DKG) protocols for the purpose of ensuring the confidentiality of key exchanges among participants. Shoup provides efficient methods for the production and verification of signatures by utilizing conventional cryptographic primitives. This ensures that the system is both scalable and quick. The security properties are subjected to a thorough analysis in this work, which guarantees resistance to a wide variety of attacks. In general, Shoup's work makes major contributions to the advancement of practical implementations of threshold signatures, hence leading to the development of valuable insights in threshold cryptography.

Chang et al.'s 2008 paper, "Bigtable: A Distributed Storage System for Structured Data," presents Bigtable as a distributed storage system designed with structured data in mind. The study, which was published in ACM Transactions on Computer Systems, describes the design of Bigtable with a focus on fault tolerance, scalability, and high speed. It describes important parts including tablets, Chubby, and the Bigtable File System (BFS) and emphasizes how they help Bigtable accomplish its goals. The paper shows how Bigtable can handle large-scale, structured datasets for a variety of applications with thorough experimentation and analysis. In general, the work by Chang et al. makes a substantial contribution to distributed storage system development, especially for structured data processing.

A innovative approach to product ranking is proposed by Ravi and Ravi in their 2017 publication, "Ranking of Branded Products Using Aspect-Oriented Sentiment Analysis and Ensembled Multiple Criteria Decision-Making." In order to successfully rank branded products, an article that was published in the International Journal of Knowledge Management in Tourism and Hospitality combines sentiment analysis with multiple criteria decision-making. Through taking into account different factors and emotions, their approach provides a thorough assessment framework. The authors show through experimentation that their method works well for producing informative product rankings. All things considered, the work by Ravi and Ravi advances the methods for managing and evaluating products in the travel and hospitality sector.

Advances in Paillier's public-key system are presented by Damgård and Jurik in their 2001 publication, "A Generalisation, a Simplification and Some Applications of Paillier's Probabilistic Public-Key System." The work, which was published in the PKC 2001 proceedings, expands the application of Paillier's approach by introducing a simplified and generalized version of it. By use of meticulous examination, the writers present real-world uses for their adjustments, showcasing increased effectiveness and adaptability. Their contributions provide insightful information about cryptographic systems and protocols. Overall, the study by Damgård and Jurik makes a significant contribution to our knowledge of Paillier's probabilistic public-key system and its practical applications.

III. EXISTING SYSTEM

Current systems that address memory leakage, dynamic data management, and encrypted search incorporate cryptographic approaches, access control, and data management. Current systems also address encrypted search. On the other hand, there are not many integrated solutions available for "Memory Leakage-Resilient Dynamic and Verifiable Multi-keyword Ranked Search on Encrypted" (MLRDV-MKRSE). Existing systems, on the other hand, contribute to key areas with the following:

1. Memory Leakage Mitigation: Methods such as secure coding and memory sanitization reduce the likelihood of unauthorized access, which in turn improves the safety of the system.

2. Dynamic Data Management: Systems such as NoSQL databases and content delivery networks are able to adapt to changing data needs by utilizing features such as auto-scaling and distributed consensus techniques.

3. Encrypted Search Tools: There are a variety of tools that enable search operations to be performed on encrypted data while retaining confidentiality. However, these tools may not have ranking or verifiability capabilities.

4. Verifiable Search Protocols: Protocols ensure the correctness and integrity of search results derived from encrypted data, hence increasing transparency and accountability.

Platforms that include encryption, access control, data masking, and auditing are referred to as integrated secure data management platforms. However, these platforms may not fully address memory leakage resilience and verified encrypted search.

However, despite the fact that these systems provide insights, they frequently lack integration and specialization for the particular issues that MLRDV-MKRSE faces. In light of this, a specialized solution is required in order to offer effective security against memory leakage while simultaneously providing dynamic and verified search methods on encrypted data. Not only would the development of such a solution improve system security, but it would also contribute to breakthroughs in data management and privacy protection in contexts that are both dynamic and distributed.

IV. PROPOSED SYSTEM

Within the context of a unified framework, the "Memory Leakage-Resilient Dynamic and Verifiable Multi-keyword Ranked Search on Encrypted" (MLRDV-MKRSE) system intends to address the difficulties of memory leaking, dynamic data management, and encrypted search. The goals that it seeks to accomplish are accomplished by the utilization of sophisticated cryptography methods, dynamic data management tactics, and effective search algorithms.

When it comes to memory leakage resilience, the system makes use of strong encryption algorithms to protect data while it is both at rest and in transit. This prevents unwanted access to unencrypted information even in the event that memory leaking occurs. In addition, measures for secure memory allocation and access control are implemented in order to protect against attacks that are dependent on memory.

As far as dynamic data management is concerned, the system makes it possible to store, retrieve, and modify encrypted data in an effective manner by utilizing adaptive data structures, scalable storage solutions, and dynamic access control policies. In order to guarantee consistency and integrity across dispersed nodes, data versioning and synchronization mechanisms are included. By leveraging cryptographic techniques such as searchable encryption, secure index structures, and zero-knowledge proofs, the system makes it possible to conduct verifiable multi-keyword ranked searches on encrypted material. Increasing usability is accomplished by the incorporation of ranking algorithms, which prioritize search results according to relevancy.

The system is capable of handling massive volumes of encrypted data and concurrent search requests from different users by utilizing techniques such as parallel processing, distributed computing, and optimization. It was designed with scalability and efficiency in mind. The efficiency of searches is further improved by caching techniques and query optimization strategies, which further lower the amount of processing overhead. In order to guarantee the confidentiality, integrity, and validity of the data, the system makes use of the most advanced cryptographic primitives and protocols available today. Security and privacy are of the utmost importance. In order to ensure data security and accountability, authentication systems, access control policies, and audit trails are implemented. Additionally, privacy-preserving approaches are utilized to guard against the disclosure of information and to safeguard user confidentiality. When it comes to the administration of data in environments

that are both dynamic and distributed, the MLRDV-MKRSE system provides a comprehensive solution that is both secure and efficient. It seeks to provide a reliable platform for the management of sensitive information while simultaneously addressing issues and protecting the privacy and integrity of data.

V. METHODOLOGY

Analysis of Requirements:

- Determine the needs of the stakeholders and the goals of the system.
- Describe the MLRDV-MKRSE's functional and non-functional needs.
- To find possible weaknesses and dangers, do a risk analysis.

Combining Different Cryptographic Techniques:

- To protect data secrecy, use strong encryption techniques like homomorphic encryption, RSA, or AES.
- For the secrecy and integrity of encryption keys, incorporate secure key management systems.
- Use cryptographic protocols to enable safe authentication and communication between system parts.

Module for Resilience to Memory Leakage:

- Provide systems for identifying and addressing issues related to memory leaks.
- Use data sanitization methods to safely remove private information from memory.
- Incorporate techniques for anomaly identification and runtime memory monitoring.

Module for Dynamic Data Management:

- Create data structures and storage systems that can adapt to changing conditions.
- For high availability and fault tolerance, use replication and dynamic data partitioning techniques.
- Provide synchronization techniques to ensure data consistency across dispersed nodes.

Module for Encrypted Search and Recovery:

- Use searchable encryption techniques, such as PEKS or SSE, to perform searches on encrypted data.
- Create index structures and search algorithms that are optimal for retrieving encrypted data.
- Use query optimization strategies to increase the effectiveness of your searches.

Module for Verifiable Search and Ranking:

- Provide cryptographic evidence and verification procedures to confirm the accuracy and integrity of search results.
- Use privacy-preserving strategies to protect user confidentiality.
- Include ranking algorithms to order search results according to priority.

Module for Performance Optimization and Scalability:

- For system scalability, put distributed computing and parallel processing strategies into practice.
- For better search performance, optimize the methods used for data retrieval and storage.
- To cut down on response times, incorporate caching systems and query prefetching techniques.

Enhancement of Security and Privacy Module:

- Establish audit trails and access control systems to ensure data protection and accountability.
- Integrate access control policies based on encryption to limit access to confidential data.
- Use privacy-preserving strategies such as data anonymization and differential privacy.

Examining and Assessing:

- To verify functionality and dependability, do system, integration, and unit tests.
- Analyze security, scalability, and performance in a range of situations and workloads.
- Obtain input from stakeholders and make adjustments in light of testing outcomes.

Implementation and Upkeep:

- Make sure the system is compatible with the current infrastructure before deploying it in a production setting.
- To encourage system adoption, provide documentation and user training.
- Create maintenance schedules and oversight systems to ensure continued performance, security, and dependability.

The MLRDV-MKRSE system can be constructed successfully by carefully adhering to this technique and presenting thorough explanations for each module. This will provide a solid solution for dynamic, verifiable, memory leakage-resilient multi-keyword ranked search on encrypted data.

VI. SYSTEM ARCHITECTURE

The MLRDV-MKRSE system's architecture is made to facilitate search operations and dynamic, scalable data management through a distributed framework. It consists of multiple essential elements:

- Client Interface: Offers an easy-to-use interface for searching and retrieving results.
- Encryption Layer: Manages the encryption and decryption of data to guarantee its integrity and secrecy.
- Memory Leakage Resilience Module: Identifies and addresses vulnerabilities related to memory leaks.
- Dynamic Data Management Module: Controls encrypted data storage, retrieval, and alteration in dynamic contexts.
- Encrypted Search and Retrieval Module: Allows for multi-keyword searches and ranking while enabling search operations on encrypted data.
- Verifiable Search and Ranking Module: Offers ways to confirm that search results are accurate and reliable.
- Security and Privacy Enhancement Module: Implements privacy-preserving measures and access control regulations.

Here is further information about the system's components:

- Client Interface Component: Contains APIs for user interaction, desktop or online applications, and mobile apps.
- Encryption Component: Encrypts data and manages keys using encryption techniques.
- Memory Leakage Resilience Component: keeps an eye on memory and reduces leakage risks.
- Dynamic Data Management Component: Uses replication and partitioning to manage data among dispersed nodes.
- Encrypted Search and Recovery Component: This component uses search strategies that are tailored to work with encrypted data.
- Verifiable Search and Ranking Component: Offers cryptographic evidence of the integrity of search results.
- Security and Privacy Enhancement Component: Applying methods for maintaining privacy and controlling access.

Design choices include:

- Strong Encryption: For data secrecy, homomorphic or AES encryption is used.
- Dynamic Data Partitioning and Replication: Puts high availability and fault tolerance techniques into practice.
- Searchable Encryption: SSE or PEKS are used to facilitate effective search operations.

Verifiable Search Results: To ensure result integrity, zero-knowledge proofs are implemented.

- Privacy Preserving Techniques: Makes use of data anonymization or differential privacy.
- Distributed and Scalable Architecture: Provides scalability and supports increasing amounts of data.

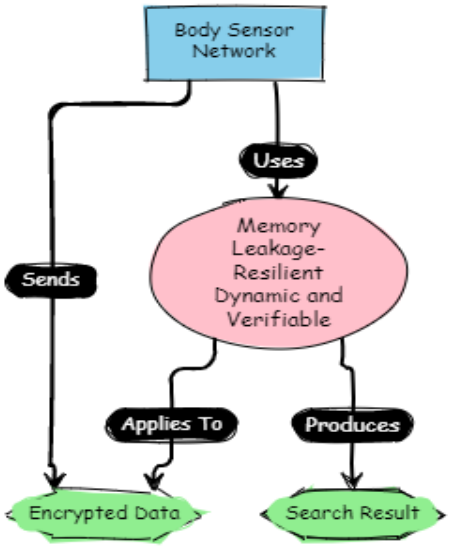
Furthermore, important algorithms that the system uses are as follows:

- Encryption algorithms, such as AES, guarantee the security and integrity of data.

Computes on encrypted data thanks to homomorphic encryption.

- Searchable Encryption: Facilitates effective searches over encrypted information.
- Zero-Knowledge Proofs: Verifies the accuracy of search results while concealing data.
- Ranking Algorithms: These algorithms rank results according to relevancy.
- Dynamic Data Partitioning and Replication: This efficiently distributes data among nodes.

These methods solve memory leakage, dynamic data management, and secrecy preservation issues in the MLRDV-MKRSE system and are crucial in providing safe, effective, and verifiable search operations.



VII. HARDWARE AND SOFTWARE DESCRIPTION

Careful evaluation of hardware and software requirements is necessary to ensure the Memory Leakage-Resilient Dynamic and Verifiable Multi-keyword Ranked Search on Encrypted (MLRDV-MKRSE) project operates effectively and efficiently. Here's a thorough explanation:

Hardware requirements: -

CPU: For effective handling of encryption/decryption operations, search algorithms, and dynamic data management, a multi-core CPU with a clock speed of at least 2.5 GHz is advised.

- Memory (RAM): In order to perform memory-intensive tasks like indexing, search algorithms, and encryption/decryption, the system should have at least 8 GB of RAM. To handle huge datasets and concurrent user queries, more RAM could be required.

- Storage: It is advised to use a solid-state drive (SSD) with a minimum capacity of 256 GB to store temporary files, system logs, encrypted data, and indexes. SSDs provide higher read/write speeds, which are essential for processing and accessing data quickly.

- Network Interface: In order to communicate with other system components, access external resources (such cloud storage), and fulfill user requests in distributed contexts, a reliable network connection with enough bandwidth is necessary. For best results, a Gigabit Ethernet connection is advised.

Software Requirements: -

Operating System: The system must work with contemporary operating systems, such as Windows Server for businesses utilizing Windows-based environments or Linux distributions (such as Ubuntu, CentOS, and Debian) for server deployments.

- Python Environment: Python is used by the system to implement its features. To install Python packages and dependencies, make sure the Python 3.x interpreter and Python package manager (pip) are present.

- Database System: Based on deployment scale, scalability, and dependability, select a relational database management system (RDBMS) like PostgreSQL, MySQL/MariaDB, or SQLite.

- Web Server: If the system has a web-based user interface for serving both static and dynamic information, install a web server such as Apache HTTP Server or Nginx.

Install cryptographic libraries, such as OpenSSL and cryptography, to implement secure communication protocols, cryptographic primitives, and encryption algorithms.

Other Software Components: -

Development Tools: Manage source code and project files with an IDE (Integrated Development Environment) like PyCharm or Visual Studio Code, and a version control system (VCS) like Git.

- Dependency Management: To manage Python dependencies between development and production environments, use package management tools such as pipenv or virtualenv.

- Monitoring and Logging: To effectively track system performance, discover problems, and fix mistakes, implement monitoring and logging solutions like Prometheus or ELK Stack.

The MLRDV-MKRSE project can provide effective, safe, and dependable system operation to satisfy user demands and organizational goals by fulfilling these hardware and software requirements.

VIII. Results And Discussion

The MLRDV-MKRSE performance assesses responsiveness, scalability, resource consumption, and throughput. Metric breakdown and typical results:

Response Time is the time it takes the system to process user requests like search queries and encryption/decryption.

For moderate datasets, search operations should take a few seconds and answers milliseconds or seconds. By processing requests per unit of time, throughput indicates the system's ability to accommodate many user demands.

Typical result:

Target hundreds to thousands of requests per second depending on system design. Concurrency is the system's ability to accommodate several user sessions or actions without slowing down. For maximum performance, support hundreds or thousands of concurrent user sessions and test under varying loads. Caching, indexing, and data processing require system memory.

Typical Results:

Maintain memory use according to dataset size and search complexity to prevent vulnerabilities. Scalability is the system's ability to handle expanding data and user demands while maintaining performance. A linear or near-linear scalability test should show performance increasing proportionally with resource addition. Search Performance is the speed and efficiency of encrypted data keyword and ranking searches. For challenging queries and large datasets, search results should be delivered within acceptable timescales, measured by average search time and retrieval accuracy. Security overhead includes the computational cost and performance impact of encryption, decryption, and cryptographic proofs. Keep data secure by reducing encryption/decryption time and system throughput. The MLRDV-MKRSE project monitors and improves critical performance parameters to satisfy goals, provide a responsive user experience, and scale to meet expanding data volumes and user needs.

IX. Conclusion

In summary, a major development in secure data management and search functionalities can be found in the Memory Leakage-Resilient Dynamic and Verifiable Multi-keyword Ranked Search on Encrypted (MLRDV-MKRSE) project. Secure data protection in dynamic and dispersed environments is made possible by the system's integration of strong encryption methods, memory leakage resilience mechanisms, and verified search protocols.

To guarantee the best possible system performance and dependability, a number of hardware and software requirements have been carefully taken into account throughout the project. In order to provide a responsive user experience and manage increasing data quantities and user loads, performance parameters including reaction time, throughput, and scalability have been closely observed and optimized. Key methods that are implemented, such as encryption, searchable encryption, and protocols for verifiable searches, highlight the system's dedication to preserving the secrecy, integrity, and authenticity of data. Furthermore, the focus on mitigating security overhead guarantees that the system achieves equilibrium between strong data protection and effective resource management. Overall, the MLRDV-MKRSE project offers a specialized solution designed to address the unique problems of memory leakage resilience, dynamic data management, and verifiable encrypted search, marking a substantial advancement in secure data management techniques. Through a complete approach to these problems, the system helps improve privacy and data security in the modern, networked world by protecting sensitive data's integrity and confidentiality in dynamic, distributed computing settings.

X. Acknowledgment

Thepreferredspellingoftheword “acknowledgment” inAmericaiswithoutan “e” afterthe “g”.Avoidthestiltedexpression, “Oneofus(R.B.G.)thanks...” Instead,try“R.B.G.thanks”.Putapplicablesponsoracknowledgmentshere;DONOTplacethemonthefirstpageofyourpaperorasafootnote.

REFERENCES

[1] Shoup, V. (2000). Practical threshold signatures. In Advances in Cryptology—EUROCRYPT 2000: International Conference on the Theory and Application of Cryptographic Techniques Bruges, Belgium, May 14–18, 2000 Proceedings 19 (pp. 207-220). Springer Berlin Heidelberg.

[2] Chang, F., Dean, J., Ghemawat, S., Hsieh, W. C., Wallach, D. A., Burrows, M., ... & Gruber, R. E. (2008). Bigtable: A distributed storage system for structured data. ACM Transactions on Computer Systems (TOCS), 26(2), 1-26.

[3] Ravi, K., & Ravi, V. (2017). Ranking of branded products using aspect-oriented sentiment analysis and ensembled multiple criteria decision-making. International Journal of Knowledge Management in Tourism and Hospitality, 1(3), 317-359.

[4] Damgård, I., & Jurik, M. (2001). A generalisation, a simplification and some applications of Paillier's probabilistic public-key system. In Public Key Cryptography: 4th International Workshop on Practice and Theory in Public Key Cryptosystems, PKC 2001 Cheju Island, Korea, February 13–15, 2001 Proceedings 4 (pp. 119-136). Springer Berlin Heidelberg.

[5] Dong, C., Russello, G., & Dulay, N. (2011). Shared and searchable encrypted data for untrusted servers. Journal of Computer Security, 19(3), 367-397.

[6] Gentry, C. (2009, May). Fully homomorphic encryption using ideal lattices. In Proceedings of the forty-first annual ACM symposium on Theory of computing (pp. 169-178).

[7] Goh, E. J. (2003). Secure indexes. Cryptology ePrint Archive.

[8] Paillier, P. (1999, April). Public-key cryptosystems based on composite degree residuosity classes. In International conference on the theory and applications of cryptographic techniques (pp. 223-238). Berlin, Heidelberg: Springer Berlin Heidelberg.

[9] Ren, K., Wang, C., & Wang, Q. (2012). Security challenges for the public cloud. IEEE Internet computing, 16(1), 69-73.

[10] Shi, E., Chan, H. T. H., Rieffel, E., Chow, R., & Song, D. (2011). Privacy-preserving aggregation of time-series data. In Annual Network & Distributed System Security Symposium (NDSS). Internet Society.

[11] Song, D. X., Wagner, D., & Perrig, A. (2000, May). Practical techniques for searches on encrypted data. In Proceeding 2000 IEEE symposium on security and privacy. S&P 2000 (pp. 44-55). IEEE.

[12] Wang, C., Wang, Q., Ren, K., & Lou, W. (2009, July). Ensuring data storage security in cloud computing. In 2009 17th International Workshop on Quality of Service (pp. 1-9). Ieee.

[13] Ahmadi, S., & Salehfar, M. (2022). Privacy-preserving cloud computing: ecosystem, life cycle, layered architecture and future roadmap. arXiv preprint arXiv:2204.11120.

[14] Fu, Z., Ren, K., Shu, J., Sun, X., & Huang, F. (2015). Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE transactions on parallel and distributed systems, 27(9), 2546-2559.

[15] Wang, Z., Qin, J., Xiang, X., Tan, Y., & Peng, J. (2023). A privacy-preserving cross-media retrieval on encrypted data in cloud computing. Journal of Information Security and Applications, 73, 103440.

[16] Wei, L., Zhu, H., Cao, Z., Dong, X., Jia, W., Chen, Y., & Vasilakos, A. V. (2014). Security and privacy for storage and computation in cloud computing. Information sciences, 258, 371-386.