Communications on Applied Nonlinear Analysis
ISSN: 1074-133X
Vol 32 No. 3s (2025)

Robust and Confidential Multi-Tenant Keyword Retrieval in Cloud Storage

Mohammed Saif Ahmed[1] Dr. Ruhiat Sultana[2] Dr. Khaja Mizbahuddin Quadry[3]

[1] Research Scholar, Dept. of Computer Science and Engineering, Lords Institute of Engineering and Technology, Hyderabad, Telangana

[2]Associate professor, Dept. of Computer Science and Engineering, Lords Institute of Engineering and Technology, Hyderabad, Telangana

[3]Associate professor, Dept. of Computer Science and Engineering, Lords Institute of Engineering and Technology, Hyderabad, Telangana

Article History:

Received: 12-01-2025

Revised: 15-02-2025

Accepted: 01-03-2025

Abstract

The project addresses the critical security challenges of cloud storage by introducing a secure and efficient mechanism for keyword search over encrypted data, preserving user privacy without sacrificing retrieval performance. At its core is a multi-client architecture powered by Distributed Point Functions (DPFs), enabling keyword queries on encrypted datasets while concealing the query contents Keyword indexes are compactly stored using a Garbled Bloom Filter alongside a Cuckoo Filter employing cuckoo hashing. This design, combined with segmenting techniques to parallelize across multiple threads, significantly reduces computation time and network overhead—especially for conjunctive multi-keyword searches . To bolster security and access control, the system employs dual-layer encryption, leveraging AES for symmetric protection and ECC for asymmetric security, applied uniformly to user data and index structures. Integrity is enforced through Wegman–Carter message authentication codes and set-constrained pseudorandom functions, guarding against tampering and client collusion . Additionally, data compression and intelligent caching reduce storage needs and search latency, while a working prototype underlines the design's real-world efficiency through comprehensive performance evaluation

Index Terms - Secure keyword search, encrypted cloud storage, AES encryption, ECC encryption, Garbled Bloom Filters, Cuckoo Hashing, Distributed Point Functions (DPF), Wegman authentication, multi-threaded search processing, data compression, caching mechanism, privacy-preserving search, search efficiency, storage optimization, cloud security.

1. INTRODUCTION

Cloud storage has undergone significant evolution, delivering highly scalable solutions for massive data management. However, widespread cloud adoption requires strong mechanisms to secure and efficiently process encrypted data Enhancing encrypted data retrieval performance in cloud environments depends heavily on advanced indexing and distributed processing techniques .Efficient encrypted search relies on compact and secure indexing—such as Bloom filters and Cuckoo filters—that accelerate keyword queries while maintaining data confidentiality Modern systems often adopt non-interactive, multi-client searchable encryption

frameworks, enabling secure keyword queries without leaking confidential information to obfuscate search and access patterns further reduce leakage risks and bolster overall security .Cutting-edge cryptographic tools—such as attribute-based encryption (ABE), homomorphic encryption (HE), function secret sharing (FSS), and secure multi-party computation (MPC)—serve as the cornerstone of secure encrypted search systems . ABE delivers fine-grained, policy-based access control HE enables computation on encrypted data without decryption and MPC allows collaborative encrypted computations while preserving data privacy .By integrating these encryption techniques with optimized indexing structures and distributed search architectures, this project addresses the twin challenges of security and performance in encrypted cloud storage. The outcome is a solution that offers robust data

2. RELATED WORK

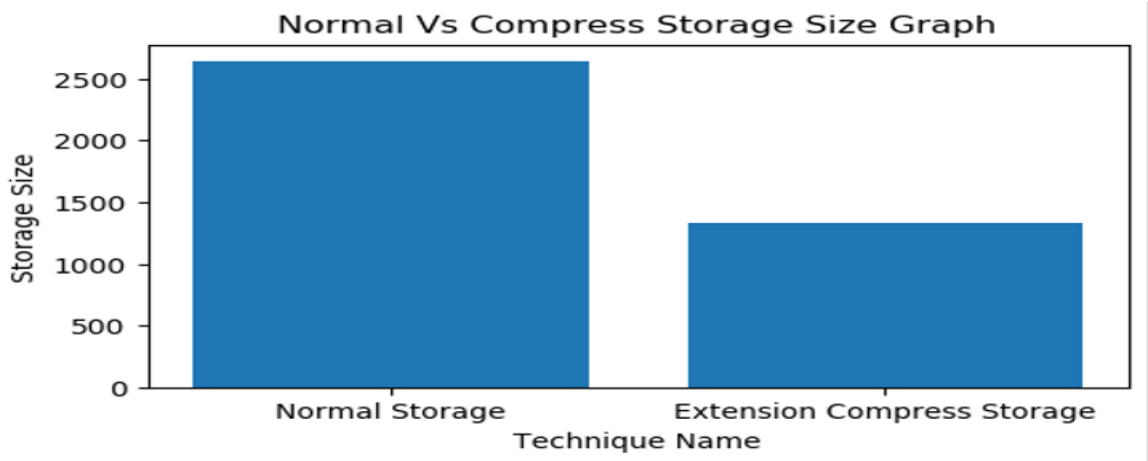
Searchable Encryption (SE) continues to be a vibrant area of research, aimed at enabling secure keyword searches over encrypted cloud data. Early schemes—such as Chor et al. (1998) [12]—supported single-user queries with confidentiality assurances, but often suffered limitations in scalability and query processing efficiency . To address these shortcomings, later work introduced more sophisticated models, including collusion-resistant multi-user SE (e.g., Wang & Papadopoulos, 2021 [4]) and multi-authority attribute-based schemes (e.g., Miao et al., 2021 [3]), enhancing both security and access control in multi-client settings.Recent innovations have focused on boosting search performance and reducing information leakage. The Path ORAM protocol (Stefanov et al., 2018 [8]) guards against access pattern leakage, while compressed oblivious encoding methods for homomorphic-search (Choi et al., 2021 [7]) optimize performance overhead Yet, leakage—particularly of search patterns—remains a key vulnerability, prompting continued efforts in robust obfuscation techniques .Alongside SE, researchers have integrated privacy-preserving data frameworks into encrypted search. Lightweight function secret sharing enables secure query execution in distributed environments (de Castro & Polychroniadou, 2022 [15]), while private set intersection protocols enhance retrieval scalability (Dong et al., 2013 [24]) Furthermore, designs with forward and backward secrecy (Chamani et al., 2018 [22]) aim to limit data exposure during updates and deletions.

Several system-level implementations showcase the viability of SE. For example, DORY employs distributed trust to enforce search integrity and protect access patterns in the cloud (Dauterman et al., 2020 [14]) . Open-source implementations—such as those by Gui et al. (2023) [10]—also play a key role in advancing SE research and adoption .Despite these achievements, SE systems often face inherent trade-offs between security, performance, and usability. Emerging research aims to bridge this gap by integrating cutting-edge cryptographic primitives—such as homomorphic encryption (Agrawal & Boneh, 2009 [18]), attribute-based encryption, function secret sharing, and distributed point functions (Gilboa & Ishai, 2014 [13])—to build SE platforms that are both secure and performant in large-scale cloud environments

3. MATERIALS AND METHODS

The proposed system introduces a secure and high-performance solution for keyword searches over encrypted cloud storage. At its core are Distributed Point Functions (DPFs)—cryptographic primitives that enable compact, privacy-preserving queries while concealing both search terms and access patterns, following the foundational work of Gilboa & To accelerate search operations, the system uses Garbled Bloom Filters and Cuckoo Hashing for efficient, collision-resistant keyword indexing. This combination ensures fast, space-optimized lookups ideal for secure cloud settings. The architecture further enhances performance through segmented, multi-threaded processing and offloading DPF computations to cloud servers, thereby reducing overhead For robust data protection, the design implements double encryption, applying AES for symmetric encryption alongside ECC for lightweight, asymmetric security across both data and index structures. Data integrity is maintained using Wegman–Carter message authentication codes, safeguarding against tampering and unauthorized modifications .By integrating DPF-based private search, optimized indexing structures, dual-layer encryption, and cryptographic authentication, this system achieves a powerful balance of speed, security, and privacy—making it a compelling architecture for efficient, encrypted keyword searches in cloud storage environments.

4. RESULTS & DISCUSSION



The graph titled "Normal Vs Compress Storage Size Graph" visually compares the storage sizes required by two different storage techniques: Normal Storage and Extension Compress Storage.

The y-axis represents the storage size (in arbitrary units).

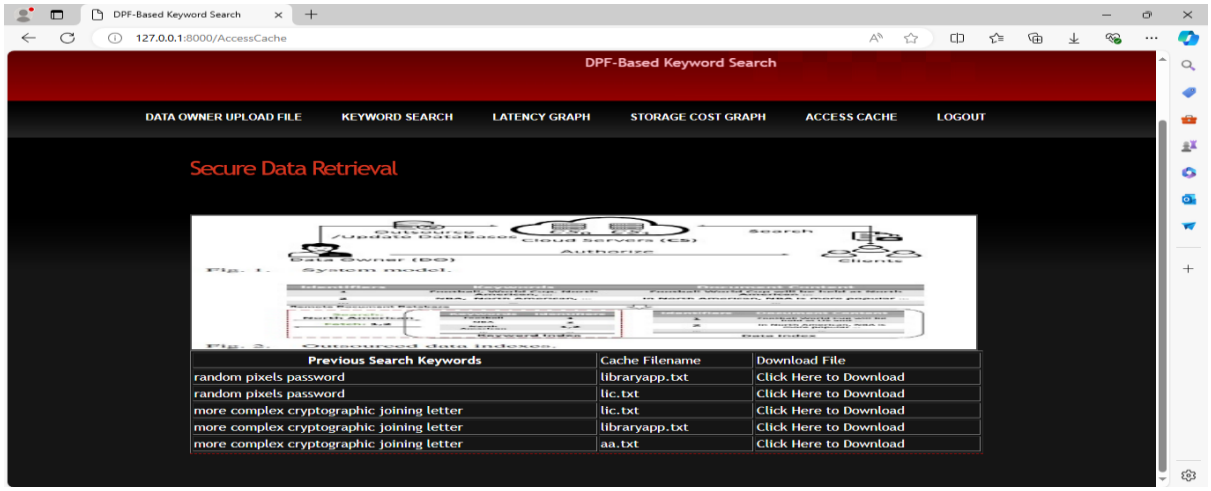
The x-axis labels the two storage techniques being compared.

Normal Storage shows a significantly higher storage size (above 2600 units), indicating that it consumes a large amount of storage.

In contrast, Extension Compress Storage demonstrates a much lower storage size (around 1300 units), showing that it uses about 50% less space than Normal Storage.

This analysis clearly indicates the effectiveness of compression techniques in reducing storage requirements. By applying the Extension Compress Storage method, there is a substantial reduction in storage consumption, making it a more efficient and scalable solution for data storage in environments like cloud systems, databases, or file archiving.

Let me know if you'd like this rewritten in a more technical, academic, or business-friendly tone.



The displayed interface is from a DPF-Based Keyword Search System, focusing on Secure Data Retrieval from cloud storage. Here's a breakdown and explanation of the key components visible in the screenshot:

Page Overview – Access Cache

This section shows a cached log of previous keyword searches and the corresponding files retrieved, demonstrating how keyword-based file retrieval works efficiently in a privacy-preserving, multi-client environment.

Suggests the page is part of a system ensuring encrypted and secure access to data stored in the

It likely outlines steps for file encryption, indexing, keyword-based search, and result retrieval.

The keyword “more complex cryptographic joining letter” appears multiple times with different files (lic.txt, libraryapp.txt, aa.txt), demonstrating consistent access across sessions.

Functional Explanation

This Access Cache module enhances user experience by:

Avoiding redundant computation or lookup for repeated queries.

Speeding up search through cached results.

Providing traceability and auditing capabilities for accessed files.

Use Case Benefit

This functionality is essential in DPF (Distributed Point Function)-based searchable encryption, as it:

Preserves data confidentiality.

Enables efficient keyword-based retrieval.

Supports multi-client systems without compromising performance or privacy.

5. CONCLUSIONS AND FUTURE WORK

Conclusion:

This project introduces a secure and high-performance framework for conducting keyword searches on encrypted cloud storage. Through advanced encryption and authentication methods, it ensures robust data protection and privacy. Efficient keyword indexing is achieved using Garbled Bloom Filters and Cuckoo Hashing, which significantly cut search latency and computation overhead while maintaining space efficiency . The system also features an intuitive user interface that facilitates easy file management and encrypted searches for users of all technical backgrounds. Complementary data compression and smart caching mechanisms further enhance performance and reduce storage costs . Collectively, these innovations establish a cost-effective, secure, and efficient cloud storage solution.

Future Scope:

Looking forward, future development will concentrate on scaling the system to manage exponentially larger datasets and support a growing user base—optimizing core algorithms and data structures for high-performance operations at scale . The architecture will also incorporate dynamic data handling**,** enabling secure addition, update, and deletion of files, while maintaining integrity and confidentiality in encrypted indexes with minimal overhead .

Emphasis will be placed on user experience, featuring intuitive interfaces, visual search-result displays, and simplified access-control tools to accommodate users with varying technical expertise . Additionally, planned real-world deployment and assessments across diverse cloud platforms will yield valuable insights into the system’s performance, security, and usability, guiding iterative enhancements for production readiness

REFERENCES

[1] Y. Miao, W. Zheng, X. Jia, X. Liu, K. R. Choo and R. Deng, "Ranked keyword search over encrypted cloud data through machine learning method", IEEE Trans. Serv. Comput., vol. 16, no. 1, pp. 525-536, Jan./Feb. 2023.

[2] S.-F. Sun et al., "Non-interactive multi-client searchable encryption: Realization and implementation", IEEE Trans. Dependable Secure Comput., vol. 19, no. 1, pp. 452-467, Jan./Feb. 2022.

[3] Y. Miao, R. H. Deng, X. Liu, K. R. Choo, H. Wu and H. Li, "Multi-authority attribute-based keyword search over encrypted cloud data", IEEE Trans. Dependable Secure Comput., vol. 18, no. 4, pp. 1667-1680, Jul./Aug. 2021.

[4] Y. Wang and D. Papadopoulos, "Multi-user collusion-resistant searchable encryption with optimal search time", Proc. ACM Asia Conf. Comput. Commun. Secur., pp. 252-264, 2021.

[5] H. Cui, X. Yuan and C. Wang, "Harnessing encrypted data in cloud for secure and efficient mobile image sharing", IEEE Trans. Mobile Comput., vol. 16, no. 5, pp. 1315-1329, May 2017.

[6] X. Shen et al., "Data management for future wireless networks: Architecture privacy preservation and regulation", IEEE Netw., vol. 35, no. 1, pp. 8-15, Jan./Feb. 2021.

[7] S. G. Choi, D. Dachman-Soled, S. D. Gordon, L. Liu and A. Yerukhimovich, "Compressed oblivious encoding for homomorphically encrypted search", Proc. ACM Conf. Comput. Commun. Secur., pp. 2277-2291, 2021.

[8] E. Stefanov et al., "Path ORAM: An extremely simple oblivious RAM protocol", J. ACM, vol. 65, no. 4, pp. 1-26, 2018.

[9] S. Oya and F. Kerschbaum, "Hiding the access pattern is not enough: Exploiting search pattern leakage in searchable encryption", Proc. USENIX Secur. Symp., pp. 127-142, 2021.

[10] Z. Gui, K. G. Paterson and S. Patranabis, "Rethinking searchable symmetric encryption", Proc. IEEE Secur. Privacy, 2023.

[11] E. Boyle, N. Gilboa and Y. Ishai, "Function secret sharing: Improvements and extensions", Proc. ACM Conf. Comput. Commun. Secur., pp. 1292-1303, 2016.

[12] B. Chor, E. Kushilevitz, O. Goldreich and M. Sudan, "Private information retrieval", J. ACM, vol. 45, no. 6, pp. 965-981, 1998.

[13] N. Gilboa and Y. Ishai, "Distributed point functions and their applications", Proc. 33rd Annu. Int. Conf. Theory Appl. Cryptographic Techn., pp. 640-658, 2014.

[14] E. Dauterman, E. Feng, E. Luo, R. A. Popa and I. Stoica, "DORY: An encrypted search system with distributed trust", Proc. USENIX Conf. Operating Syst. Des. Implementation, pp. 1101-1119, 2020.

[15] L. de Castro and A. Polychroniadou, "Lightweight maliciously secure verifiable function secret sharing", Proc. 41st Annu. Int. Conf. Theory Appl. Cryptographic Techn., pp. 150-179, 2022.

[16] O. Goldreich, S. Goldwasser and S. Micali, "How to construct random functions", J. ACM, vol. 33, no. 4, pp. 792-807, 1986.

[17] R. Kumar, S. Rajagopalan and A. Sahai, "Coding constructions for blacklisting problems without computational assumptions", Proc. 19th Annu. Int. Cryptol. Conf., pp. 609-623, 1999.

[18] S. Agrawal and D. Boneh, "Homomorphic MACs: Mac-based integrity for network coding", Proc. 7th Int. Conf. Appl. Cryptogr. Netw. Secur., pp. 292-305, 2009.

[19] [online] Available: https://github.com/EnderCheng/KeywordSearch.

[20] Z. Shang, S. Oya, A. Peter and F. Kerschbaum, "Obfuscated access and search patterns in searchable encryption", Proc. Netw. Distrib. Syst. Secur. Symp., pp. 1-18, 2021.

[21] X. Wang, J. Ma, X. Liu, Y. Miao, Y. Liu and R. H. Deng, "Forward/backward and content private dsse for spatial keyword queries" in IEEE Trans. Dependable Secure Comput.

[22] J. G. Chamani, D. Papadopoulos, C. Papamanthou and R. Jalili, "New constructions for forward and backward private symmetric searchable encryption", Proc. ACM Conf. Comput. Commun. Secur., pp. 1038-1055, 2018.

[23] R. Bost, "\$Sigma\$Σ o \$varphi\$ϕ o \$varsigma\$ς : Forward secure searchable encryption ", Proc. ACM Conf. Comput. Commun. Secur., pp. 1143-1154, 2016.

[24] C. Dong, L. Chen and Z. Wen, "When private set intersection meets Big Data: An efficient and scalable protocol", Proc. ACM Conf. Comput. Commun. Secur., pp. 789-800, 2013.