A SECURE AND DYNAMIC MULTI-KEYWORD RANKED SEARCH SCHEME

[1] KAMPATI RAGHU NAGA ROHIT,[2]S.K.ALISHA

[1]MCA Student,B V Raju College, Bhimavaram,Andhra Pradesh,India

[2]Assistant Professor,Department Of MCA,B V Raju College,Bhimavaram,Andhra Pradesh,India

ABSTRACT

With the growing adoption of cloud computing, data owners are increasingly outsourcing their data to cloud servers for convenience and cost efficiency. However, ensuring data privacy requires encrypting sensitive information before outsourcing, which complicates traditional data retrieval methods such as keyword-based searches. To address this challenge, we propose a secure multi-keyword ranked search scheme over encrypted cloud data that also supports dynamic update operations, including document insertion and deletion. Our approach integrates the vector space model and the widely used TF-IDF model in index construction and query generation. We design a specialized tree-based index structure and introduce a Greedy Depth-first Search algorithm to enhance search efficiency. To ensure both security and accurate relevance ranking, we employ the secure kNN algorithm for encrypting index and query vectors. Additionally, phantom terms are incorporated into the index vector to counteract statistical attacks and prevent information leakage. Leveraging our tree-based index structure, the proposed scheme achieves sub-linear search complexity while efficiently handling dynamic updates. Experimental evaluations demonstrate that our method offers a secure, efficient, and scalable solution for multi-keyword ranked search over encrypted cloud data.

Keywords: Cloud Computing, Data Privacy, Encrypted Data, Multi-Keyword Search, Ranked Search, TF-IDF, Index Construction, Secure kNN, Dynamic Updates, Phantom Terms, Search Efficiency, Data Encryption, Secure Search Schem.

I.INTRODUCTION

Cloud computing has emerged as a transformative model for enterprise IT infrastructure, providing scalable computing, storage, and application resources. It enables users to access shared resources on demand with high efficiency and minimal cost. Due to these advantages, both individuals and organizations are increasingly outsourcing their data to cloud servers, reducing the need for dedicated software and hardware management.Despite its benefits, cloud

computing raises significant privacy concerns, particularly when outsourcing sensitive data such as personal emails, health records, financial information, and government documents. Cloud service providers (CSPs), which store and manage data on behalf of users, may gain unauthorized access to confidential information. To ensure data privacy, a common approach is data encryption before outsourcing. However, encryption significantly reduces data usability, making traditional plaintext-based keyword search techniques inapplicable. Retrieving relevant data by downloading and decrypting the entire dataset locally is impractical due to excessive storage and computational costs.

To address this challenge, researchers have explored various solutions, including fully homomorphic encryption (FHE) and oblivious RAM (ORAM) techniques. While these methods offer strong security guarantees, their high computational overhead limits their practical adoption. A more efficient approach is searchable encryption (SE), which enables encrypted data storage and retrieval without exposing plaintext information. Various SE schemes have been proposed, supporting functionalities such as single-keyword search, similarity search, Boolean multi-keyword search, and ranked search. Among them, multi-keyword ranked search has gained increasing attention due to its practicality in real-world applications.

Recently, dynamic searchable encryption schemes have been introduced to support document insertion and deletion. Since cloud-based data is frequently updated, dynamic search solutions are essential. However, only a few existing schemes efficiently support both multi-keyword ranked search and dynamic operations.

To bridge this gap, we propose a secure tree-based search scheme that enables multi-keyword ranked search while supporting dynamic document updates. Our method integrates the vector space model and the TF-IDF (Term Frequency–Inverse Document Frequency) model to enhance search accuracy. To optimize efficiency, we design a tree-based index structure and introduce a Greedy Depth-first Search (GDFS) algorithm, enabling sub-linear search complexity. Additionally, we employ the secure kNN algorithm to encrypt index and query vectors while maintaining accurate relevance scoring.

To counter various security threats, we develop two variants of our scheme:

1. Basic Dynamic Multi-keyword Ranked Search (BDMRS): Designed for the known ciphertext model to ensure fundamental security guarantees.

2. Enhanced Dynamic Multi-keyword Ranked Search (EDMRS): Developed for the known background model, offering stronger resistance to adversarial attacks.

Our Contributions

The key contributions of this paper are as follows:

1. Efficient Searchable Encryption: We propose a secure multi-keyword ranked search scheme that supports dynamic updates, allowing insertion and deletion of documents while maintaining search accuracy.
2. Optimized Search Efficiency: Our tree-based index structure ensures logarithmic search complexity, further enhanced by the Greedy Depth-first Search (GDFS) algorithm. Parallel search execution is also supported, significantly reducing search time.

II.PROPOSED SCHEMES

In this section, we first introduce the Unencrypted Dynamic Multi-keyword Ranked Search (UDMRS) scheme, which is built on the vector space model and the KBB tree. Based on this foundational scheme, we then construct two secure search schemes—BDMRS (Basic Dynamic Multi-keyword Ranked Search) and EDMRS (Enhanced Dynamic Multi-keyword Ranked Search)—to address two distinct threat models.

Index Construction in the UDMRS Scheme

As briefly discussed in Section 3, the KBB index tree structure serves as the basis for our index construction process. To build the index tree, we follow these steps:

1. Leaf Node Generation: Each document in the collection is represented as a tree node, which becomes a leaf node in the index tree.
2. Internal Node Generation: Internal nodes are created based on the leaf nodes, forming a hierarchical tree structure.
3. Index Tree Construction: The final index tree T is constructed in plaintext format, facilitating fast search and retrieval operations.

Data Structure and Notations

Before presenting the formal construction process, we define the tree node structure and key notations used in Algorithm 1:

1. Tree Node Structure: Each node in the index tree is defined as (ID, D, P_l, P_r, FID), where:
2. ID – A unique identifier for the node, generated using the function GenID().
3. D – A data field storing keyword-related information.

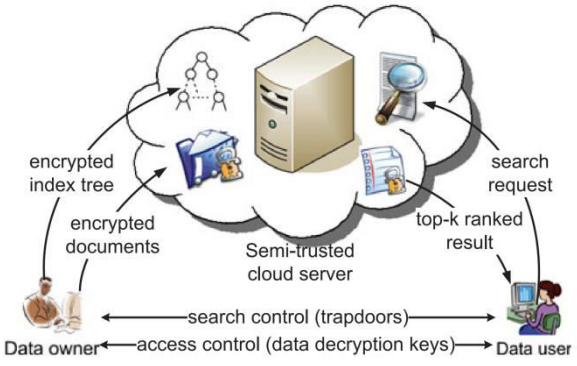
4. P_l, P_r – Pointers to the left and right child nodes.
5. FID – A field used for indexing and query matching.
6. CurrentNodeSet: The set of currently processed nodes that do not have parent nodes.
7. If the number of nodes is even, the cardinality is 2^h (where h is a positive integer).
8. If the number of nodes is odd, the cardinality is $(2^h + 1)$.
9. TempNodeSet: The set of newly generated internal nodes.

Search Process of the UDMRS Scheme

The search process in the UDMRS scheme follows a recursive procedure based on a Greedy Depth-first Search (GDFS) algorithm. The search operation utilizes a result list (RList), where each element is represented as (RScore, FID). Here, RScore denotes the relevance score of the document FID with respect to the query and is calculated using Formula (1). The RList maintains the top k documents with the highest relevance scores, ranking them in descending order based on RScore. During the search process, this list is dynamically updated to ensure that only the most relevant documents are retained.

Several key notations are used in the search algorithm. The function RScore(D_u, Q) computes the relevance score between the query vector (Q) and the index vector (D_u) stored in node u, as defined in Formula (1). The variable kthscore represents the smallest relevance score among the current top-k documents in RList and is initially set to zero. Additionally, each tree node has two child nodes: hchild, the child with the higher relevance score, and lchild, the child with the lower relevance score.

By leveraging the tree structure, the algorithm efficiently narrows the search space. Since the maximum possible relevance score of the documents rooted at a node u can be estimated in advance, only a subset of the nodes in the tree needs to be accessed. This optimization significantly reduces search time while maintaining accuracy. An example of the search process is illustrated in Figure 3, which considers a document collection $F = \{f_i \mid i = 1, ..., 6\}$, a dictionary of size m = 4, and a query vector Q = (0, 0.92, 0, 0.38).



III.CONCLUSION

In this paper, we propose a secure, efficient, and dynamic searchable encryption (SE) scheme that supports multi-keyword ranked search while enabling dynamic deletion and insertion of documents. By constructing a keyword-balanced binary tree as the index and introducing a Greedy Depth-first Search algorithm, our scheme achieves higher efficiency compared to traditional linear search methods. Additionally, parallel execution further enhances the search performance, reducing overall time complexity. The security of the scheme is ensured through the secure kNN algorithm, protecting against potential threats in the known ciphertext and known background models. Experimental evaluations demonstrate the effectiveness and efficiency of our approach.

Despite these advancements, several challenges remain in symmetric SE schemes. The current design requires the data owner to manage index updates and store unencrypted index trees, which may not be ideal for cloud computing environments. A promising future direction is to develop a fully cloud-based dynamic searchable encryption scheme that allows the cloud server to handle updates while maintaining multi-keyword ranked search functionality. Additionally, multi-user security poses further challenges. User revocation remains complex, as it requires index reconstruction and key redistribution. Moreover, dishonest data users may share decrypted documents or distribute secure keys to unauthorized individuals, creating security vulnerabilities.

Future work will focus on addressing these challenges by enhancing security mechanisms, improving user revocation processes, and developing a more robust and practical SE scheme that effectively balances efficiency, security, and usability in multi-user cloud environments.

IV.REFERENCES

1. K. Ren, C. Wang, and Q. Wang, "Security challenges for the public cloud," *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, Jan.–Feb. 2012.

2. S. Kamara and K. Lauter, "Cryptographic cloud storage," in *Proceedings of Financial Cryptography and Data Security*, 2010, pp. 136–149.

3. C. Gentry, *A Fully Homomorphic Encryption Scheme*, Ph.D. dissertation, Stanford University, Stanford, CA, USA, 2009.

4. O. Goldreich and R. Ostrovsky, "Software protection and simulation on oblivious RAMs," *Journal of the ACM*, vol. 43, no. 3, pp. 431–473, 1996.

5. D. Boneh, G. Di Crescenzo, R. Ostrovsky, and G. Persiano, "Public key encryption with keyword search," in *Proceedings of Advances in Cryptology – EUROCRYPT*, 2004, pp. 506–522.

6. D. Boneh, E. Kushilevitz, R. Ostrovsky, and W. E. Skeith III, "Public key encryption that allows PIR queries," in *Proceedings of Advances in Cryptology*, 2007, pp. 50–67.

7. D. X. Song, D. Wagner, and A. Perrig, "Practical techniques for searches on encrypted data," in *Proceedings of the IEEE Symposium on Security and Privacy*, 2000, pp. 44–55.

8. E.-J. Goh, "Secure indexes," *IACR Cryptology ePrint Archive*, vol. 2003, p. 216, 2003.

9. Y.-C. Chang and M. Mitzenmacher, "Privacy-preserving keyword searches on remote encrypted data," in *Proceedings of the 3rd International Conference on Applied Cryptography and Network Security*, 2005, pp. 442–455.

10. R. Curtmola, J. Garay, S. Kamara, and R. Ostrovsky, "Searchable symmetric encryption: Improved definitions and efficient constructions," in *Proceedings of the 13th ACM Conference on Computer and Communications Security*, 2006, pp. 79–88.

11. 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*, 2010, pp. 1–5.

12. M. Kuzu, M. S. Islam, and M. Kantarcioglu, "Efficient similarity search over encrypted data," in *Proceedings of the IEEE 28th International Conference on Data Engineering*, 2012, pp. 1156–1167.

13. C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in *Proceedings of IEEE INFOCOM*, 2012, pp. 451–459.

14. B. Wang, S. Yu, W. Lou, and Y. T. Hou, "Privacy-preserving multi-keyword fuzzy search over encrypted data in the cloud," in *Proceedings of IEEE INFOCOM*, 2014, pp. 2112–2120.

15. P. Golle, J. Staddon, and B. Waters, "Secure conjunctive keyword search over encrypted data," in *Proceedings of Applied Cryptography and Network Security*, 2004, pp. 31–45.

16. Y. H. Hwang and P. J. Lee, "Public key encryption with conjunctive keyword search and its extension to a multi-user system," in *Proceedings of the 1st International Conference on Pairing-Based Cryptography*, 2007, pp. 2–22.

17. L. Ballard, S. Kamara, and F. Monrose, "Achieving efficient conjunctive keyword searches over encrypted data," in *Proceedings of the 7th International Conference on Information and Communications Security*, 2005, pp. 414–426.

18. D. Boneh and B. Waters, "Conjunctive, subset, and range queries on encrypted data," in *Proceedings of the 4th Conference on Theory of Cryptography*, 2007, pp. 535–554.

19. B. Zhang and F. Zhang, "An efficient public key encryption with conjunctive-subset keyword search," *Journal of Network and Computer Applications*, vol. 34, no. 1, pp. 262–267, 2011.

20. J. Katz, A. Sahai, and B. Waters, "Predicate encryption supporting disjunctions, polynomial equations, and inner products," in *Proceedings of Advances in Cryptology*, 2008, pp. 146–162.

21. E. Shen, E. Shi, and B. Waters, "Predicate privacy in encryption systems," in *Proceedings of the 6th Theory of Cryptography Conference on Theory of Cryptography*, 2009, pp. 457–473.

22. A. Lewko, T. Okamoto, A. Sahai, K. Takashima, and B. Waters, "Fully secure functional encryption: Attribute-based encryption and (hierarchical) inner product encryption," in *Proceedings of the 29th Annual International Conference on Theory and Applications of Cryptographic Techniques*, 2010, pp. 62–91.

23. A. Swaminathan, Y. Mao, G.-M. Su, H. Gou, A. L. Varna, S. He, M. Wu, and D. W. Oard, "Confidentiality-preserving rank-ordered search," in *Proceedings of the ACM Workshop on Storage Security and Survivability*, 2007, pp. 7–12.

24. S. Zerr, D. Olmedilla, W. Nejdl, and W. Siberski, "Zerber+R: Top-k retrieval from a confidential index," in *Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology*, 2009, pp. 439–449.

25. C. Wang, N. Cao, K. Ren, and W. Lou, "Enabling secure and efficient ranked keyword search over outsourced cloud data," *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1467–1479, Aug. 2012.