International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. , No. (Mon-20..)

http://dx.doi.org/10.12785/ijcds/XXXXXX

Optimizing Encrypted Cloud Data Security and Searchability through Multi-Keyword Ranking Search Methods

Narendra Shyam Joshi 1[1], Kuldeep P. Sambrekar 2[2], Abhijit J. Patankar 3[3], Archana Jadhav 4[4] and Prajakta Khadkikar 5 [5]

[1]*Department of Computer Science and Engineering,KLS Gogte Institute of Technology, Affiliated to Visvesvaraya Technological University, Belagavi Karnataka, India*
[2]*Department of Computer Science and Engineering,KLS Gogte Institute of Technology, Affiliated to Visvesvaraya Technological University, Belagavi Karnataka, India*
[3]*Department of Information Technology, D Y Patil College of Engineering, Affiliated to Savitribai Phule Pune University, Akurdi Pune, India*
[4]*Department of Artificial Intelligence and Data Science, D Y Patil Institute of Engineering Management and Research, Affiliated to Savitribai Phule Pune University, Akurdi Pune, India*
[5]*Department of Computer Science and Engineering,School of Engineering and Technology DES Pune University, Pune*

Abstract: Encryption is becoming more and more crucial for protecting user privacy as cloud services gain popularity. It is essential to provide dependable methods for quick and safe data recovery. This study suggests a brand-new method for searching encrypted cloud data. The proposed technique uses a greedy depth-first search (DFS) algorithm combined with an advanced grading system to optimise queries including multiple words and synonyms. Users are assumed to search using a large number of keywords, some of which may be synonyms for article terms, according to the recommended architecture. To address this issue, a search algorithm that makes use of synonyms from user queries was developed. Greedy search techniques assist us in locating the most relevant data even if the search universe is constantly expanding. Our depth-first search approach increases the probability of discovering relevant data. Additionally, our research employs a novel ranking algorithm that evaluates a text's relevance to a search query based on keyword proximity, synonym accuracy, and frequency. In simulated cloud architecture tests employing industry-standard protocols and encrypted datasets, our proposed technique performs better than the state-of-the-art approaches. Runtime, recall, and accuracy all demonstrate this advantage. The greedy Depth-First Search (DFS) method increases efficiency by optimising resources. By automatically sorting the results, a grading technique assists users in finding the most relevant articles fast. In protected cloud storage systems, this synonym-enhanced search method may boost privacy and usability now.

Keywords: Cloud Security, Multi-Keyword Ranking, Greedy Depth-First Searching, Encrypted Data Retrieval, Synonym-Based Search Algorithms, Searchable Encryption

1. INTRODUCTION

Although data is so crucial in today's society, the quickly growing cloud environments have taken over as the main places to store vast quantities of data. But this ease also comes with a lot of drawbacks, especially when it comes to search functionality and data security. Sensitive data must be encrypted before being transported to the cloud in order to guarantee data confidentiality and speedy retrieval. [1] [2]To solve these issues concurrently, this paper presents a novel method named "Greedy Depth-First Search and Ranking for Synonym-Enhanced Multi-Keyword Search in Encrypted Cloud Environments". The volume of data being sent and kept on distant servers has significantly increased as a result of the development of cloud computing. The intricate details and privacy requirements of finding encrypted content are beyond the capabilities of traditional search techniques. This restriction is particularly noticeable when users have to do several keyword searches, which may include synonymous phrases and complicate the retrieval process. [1]The suggested method combines the adaptability of synonym-based searching with the computational efficiency of a modified greedy depth-first search (DFS) algorithm. By giving nodes precedence based on predetermined criteria, this updated DFS algorithm enables more targeted searches and increased retrieval efficiency. But when it comes to encrypted data, it necessitates the creation of fresh indexing

and search techniques that can decipher and explore the data while keeping it private. We also include synonym detection to improve the capability of searching for various keywords. This improvement respects the multiplicity of meanings attached to language and the fact that different people may use different words to refer to the same ideas. Our approach greatly increases the relevancy of search results by including a synonym recognition component, so users may access relevant content rather than simply content that precisely matches the query. Synonym recognition combined with the greedy depth-first search (DFS) algorithm yields a ranked search strategy that efficiently navigates encrypted content. This method gives consumers a collection of results that have been filtered and prioritised according to how relevant they are to the search parameters they have entered. This rating is important because it may help users find the most relevant information quickly and easily, saving them the trouble of reading through each paragraph that is returned one at a time. This introduction provides a thorough overview of our system, clarifying the basic algorithms that support its operation as well as its design principles. In this paper, we outline our system's architectural layout, demonstrate its useful use in real-world settings, and demonstrate its superiority over current approaches by thorough research and testing. Our method is a significant achievement in the field of data security and retrieval, driven by the increasing need for encrypted cloud environments that provide safe, effective, and intelligent search capabilities. We want to provide a hybrid strategy that provides the user with privacy-preserving multi-keyword search functionality, allowing them to quickly and accurately get relevant results. In light of this finding, the study's goals include formulated which are formally stated as.
To study and analyze the existing searchable encryption schemes.

- To research and evaluate the current searchable encryption technologies.
- To suggest a productive multi-keyword ranked search system that protects privacy.
- To provide a productive hybrid search strategy that protects privacy while ranking keywords using conjunctive and disjunctive queries on encrypted cloud data.
- To put into practice the suggested hybrid, conjunctive, and disjunctive multi-keyword ranked search algorithms that preserve privacy.
- To assess and contrast the effectiveness of the suggested hybrid, conjunctive, and disjunctive privacy-preserving multi-keyword ranked search systems with the current scheme.

2. Related work

Research has focused on developing secure and efficient search schemes over encrypted data. [3]This includes techniques like searchable encryption, where keywords are encrypted in such a way that it's still possible to search for them without decryption.

3. RESEACH GAP IDENTIFIED

The Examining multi-keyword ranked search systems in cloud settings exposes a number of research gaps that provide chances for further developments. Enhancing computing speed and efficiency, particularly in systems that combine access control and encryption, expanding language support for encrypted searches outside popular languages, and identifying the best solutions using algorithms like greedy and DFS are important topics for future work[9]. Additionally, adaptive frameworks are required to better manage searches in harsh environments and distributed networks they also need to balance privacy, security, and usability without making major compromises and they need to develop universal models for compound keyword searches that are scalable and efficient[10]. In order to close these gaps and advance the development of safe, effective, and user-friendly cloud-based search systems, an interdisciplinary strategy integrating computer science, cybersecurity, languages, and user experience insights is needed.

4. DESIGN GOALS

The proposed scheme should strive to fulfill the following design aspirations.

- Creative Search: The system need to be able to manage searches that include many keywords. As an alternative, it need to let users enter anywhere between two and five terms to mimic their real-world search behaviours.
- Fruitless Exploration: Users should quickly mark searches as failed in order to save them from costing large amounts of money. It is deemed a failure exploration if the search phrases are not present in any text inside the collection. It is feasible to reveal a fruitless search by making the fewest comparisons possible.
- Evaluate Retrieval: It is desirable to rank the supplied search query results according to their relevance to the query in order to reduce the computational load that post-processing places on end users.
- Optimised Efficiency: Easy access to the documents should be possible when the search process is effective. Reducing the number of comparisons between encrypted queries and indexes is one way to increase efficiency. Search efficiency, rank efficiency, and search accuracy—all of which are measured using measures like recall and precision—all contribute to the overall efficacy of search engines. In this

TABLE I. Car Database

Title	Authors	Year	Source	Key Findings	Probable Algorithms/Methods
Multi-keyword ranked search with access control for multiple data owners in the cloud	J Guo, C Tian, X Lu, L Zhao, Z Duan [4]	2024	Journal of Information Security and Applications, Elsevier	Proposes a secure multi-keyword ranked search system with access control mechanisms to enhance data security in multi-owner cloud environments.	Access control mechanisms combined with encryption for secure multi-keyword search.
A Hybrid Approach for Improving Data Security in Cloud Computing using Greedy DFS Ranked Searching [5]	NS Joshi, KP Sambrekar, J Abhijeet, S Allagi	2023	Journal of Intelligent Systems and Applications, ijisae.org	Combines greedy DFS with ranked searching to improve data security and privacy preservation in cloud computing, focusing on efficiency and effectiveness	Greedy Depth-First Search (DFS) and ranked search algorithms for data security
Multi-keyword Ranked Search Scheme Supporting Extreme Environments for the Internet of Vehicles[6]	D Xu, C Peng, W Wang, K Dev	2023	IEEE Internet of Things Journal	Develops a ranked search scheme for Internet of Vehicles in extreme environments, focusing on privacy protection and the security of the distributed RSU architecture	Search schemes tailored for distributed networks and extreme environments, possibly incorporating machine learning for optimization
Compound Keyword Level Search to conserve Privacy in access of Encrypted Cloud	P Karuppasamy, G Karthikeyan, MR Sankarganesh [7]	2023	ResearchGate	Offers a method for multi-keyword ranked search over encrypted cloud data, emphasizing bandwidth cost reduction and strict privacy preservation	Algorithms for compound keyword search and encryption techniques to ensure privacy and efficiency
Efficient Secure Privacy Preserving Multi Keywords Rank Search over Encrypted Data in Cloud Computing[8]	M Ali, H He, A Hussain, M Hussain, Y Yuan	2023	Journal of Information Security and Applications, Elsevier	Designs a secure and privacy-preserving multi-keyword rank search with access control, aiming at precise contribution and improved security in cloud computing	Secure multi-keyword rank search algorithms with enhanced privacy-preserving features and access control

section, we will explore the many stages that make up the overall plan for building the necessary search algorithms.

5. PROPOSED METHODOLOGY

The hk1, hk2,..., hkn set of classified cryptographic codes, collectively referred to as HK, is used in the index's creation. [11]Furthermore, this particular group, HK, serves as the entry point for the creation of questions. The collection of encryption keys, SK=sk1,sk2, skN, is meant to protect N texts in the collection. The data custodian is believed to be the one who conceptualises and manages these N secret keys in the suggested scheme. The data owner also aligns the texts to be revealed on the server with plaintext IR as part of the preparation step. Many technologies exist to reduce the initial processing load, including R and Apache Lucene. The messages are then decoded using appropriate decoders. After the decoding process, tokens, also known as keywords, are produced. Next, stop words are eliminated. The size of the keyword lists is decreased by removing stop words from the register of keywords. All of the tokens are changed to lowercase characters after the tokenization procedure[12]. The tokens are then run through stemming algorithms to get the root (or base) form of the tokens. A variety of stemmers, including the Porter, Paice, and Lovins stemmers, are available for stemming. To aid in the order of results (as carried out by several search engines), the relevance score of keywords (like TF, TF-IDF) is calculated. . The various steps executed during the preprocessing of texts are illustrated in Figure 2.

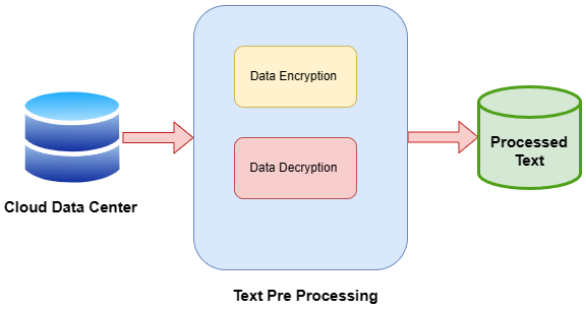

Figure 1. Enter Caption

A. Text Index Generation Phase

The DO(Data Owner) embarks on an imaginative voyage with the following stages to build the encrypted index for every text (Di, where $1 \leq i \leq N$) $\tilde{1}$:{0, 1}∗→HMACkey

Step 1: The DO uses a powerful method called HMAC for every keyword (kx) .(1) hidden in the multiple depths of text Di (assuming m keywords per text).[13] This method, which resembles a magical chant, transforms the input into an enthralling output of fixed length (l). A set of keys, called HK, was given to the DO during the setup ritual. These keys hold the power to unlock the encrypted realm's secrets, thus it is the DO's sacred duty to guard and support them.

$$H_1 : \{0,1\}^*[]HMACkey\{0,1\}^1 \tag{1}$$

Step 2: Divide the l-bit binary string into z segments, where d bits is the length of each segment.[14] Make a substring out of every segment. using the formula zj = zjd3 , zjd2 ,zjd1, zjd0 for all values of j from 1 to r.
Step 3: use to reduce the d-bit substring zj to a single bit (either 0 or 1) equation (2). The output bit for keyword kx in the keyword index Ikx [j] is 0 if all the bits in the d-bit substring zj are equal to 0 (zjd3 = 0, zjd2 = 0, zjd1 = 0, zjd0 = 0), and 1 otherwise

$$I_{kx} = [j] = \begin{cases} 0, & if\ Zjd-1 = 0\text{^}..\ \text{^}Zj2 = 0\text{^}Zj1 = 0\text{^}0 = 0 \\ 1, & \text{if otherwise} \end{cases} \tag{2}$$

In Step 4, By performing a bitwise AND operation on the indexes of the m keywords, we may determine an r-bit index (IDi) for text Di, equation (3). In order to accomplish this, we extract each individual bit from the index and utilize it as a value in a bitwise AND operation with all of the keyword indexes. By default, the value of IDi[j] is initialized to 0. However, it is changed to 1 only if all the keyword indexes have a 1 at the exact same bit position. The subsequent depiction demonstrates the execution of this procedure.

$$I_{Di}[j] = \odot_{kx=1}^{m} Ikx[j] \quad 1 \leq j \leq r \ldots\ldots \tag{3}$$

B. Clustering Text Generation Phase

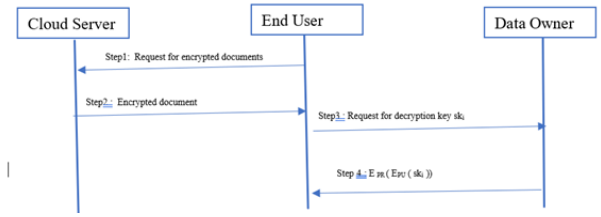

Figure 2. working of Simple data encryption and data Decryption

By using a technique to prioritize search results to give the most relevant resources, plain-text information retrieval streamlines the process for users by removing the need to wade through irrelevant content. [15]The problem with Boolean retrieval is that it floods users with incorrect content, which causes processing overhead during retrieval, decryption, and rejection.

We suggest doing the TF-IDF score calculations for the text's keywords while the process is offline[16]. To classify the keywords into different levels of relevance, we use the TF-IDF scores. Assuming their scores meet or exceed the requirements for that level, keywords can be located in both

earlier and current levels. Reliability in ranking and retrieval is guaranteed by this.

C. System Architecture

In the figure 6 to show System Architecture allows users to input multi-keyword search queries, which might include synonyms and exclusion phrases.

- **Query Pre-Processing:** Determines synonymous terms by utilising an internal or external repository of information.[17] Implements discretionary query expansion to enhance retrieval by including more results. Creates a well-organized query using weighted Boolean operators (OR, AND, NOT).
- **Query Encryption:** Utilises robust encryption methods, such as searchable encryption, to safeguard the secrecy of queries. Guarantees that encrypted queries do not disclose any sensitive information to the server.
- **Cloud Server Side:** The system stores an encrypted index of texts that can be searched using keywords, allowing retrieval of information without the need for decryption.
- **Encrypted Query Processing:** Utilises efficient search algorithms to compare encrypted requests with the encrypted index.
- **Greedy Depth-First Search (DFS) Algorithm** Traverses the encrypted index structure using a depth-first approach. [18]Assigns more priority to branches that have greater potential importance, as determined by their intermediate ratings. Assigns more priority to branches that possess greater potential significance, as judged by their intermediate grades.

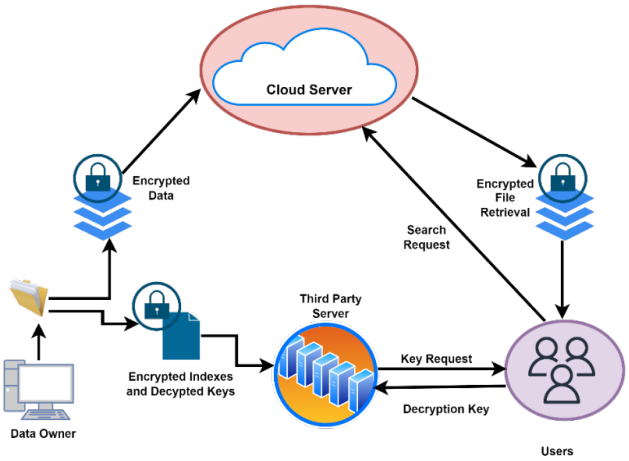

Figure 3. System Architecture

- **Ranking:** Employs the Rank(S, Q) formula to calculate relevance scores for retrieved texts: w1 * OR (Synonym1, Synonym2, ..., SynonymN) w2 * AND (Keyword1, Keyword2, ..., KeywordN) w3 * NOT (UnwantedKeyword1, UnwantedKeyword2, ..., UnwantedKeywordM)
Adjusts weights (w1, w2, w3) for desired balance between synonyms, mandatory keywords, and exclusion terms.
- **Client-Side:** Obtains ciphered outcomes from the server. Deciphers the data using the suitable encryption key.
- **Result Presentation:** Displays decrypted texts in ranked order based on calculated relevance score Here is a high-level plan for a Greedy DFS and Ranking algorithm enhanced with synonyms for use in multi-keyword searches in secure cloud storage.
- **Data Encryption:** The data undergoes encryption through a sophisticated encryption mechanism, enabling its secure storage in the cloud while preserving the capability to do searches on encrypted terms.
- **Synonym Dictionary Creation:** We develop a comprehensive synonym dictionary that maps keywords to their synonymous counterparts. [19]This is integrated into the search mechanism to enhance the search results' relevance by capturing the semantic relationships.
- **Greedy DFS Algorithm for Search:** A DFS algorithm with a greedy strategy is suggested as a means to effectively explore the search space[20]. [21]The system assigns higher priority to nodes (encrypted files) that have a higher likelihood of containing the desired keywords.[22] This prioritisation is determined by a heuristic that takes into account both the presence of the principal keywords and their synonyms.

6. METHODS AND ALGORITHMS

The equation for the heuristic might look something like this:

$$H(n)=\alpha f(n)+\beta g(n)$$

Where , **H(n)** is the heuristic function for node n.
f(n) is a function that returns a value representing the presence of the principal keywords at node n.
g(n) is a function that returns a value representing the presence of synonyms of the keywords at node n. $\alpha$and β are weighting factors that determine the relative importance of the presence of principal keywords and their synonyms, respectively.

In a greedy DFS, the heuristic value is used to determine which child receives attention next. Here is a graphical representation of the algorithm as follows.

A. Greedy DFS(node) heuristic Search Algorithm

Greedy DFS(node):
step 1: if target is present in node
step 2: Give back the node
step 3: designate the visited node
Children = get_children(node) in step four
step5: Arrange kids according to H(n), decreasing
step 6: for the kid inside the child
step 7: Should the youngster not be visited:
step 8: Greedy DFS(child) = result
step 9: Should the outcome not be None:
step 10: Provide the outcome
step 11: Give back nothing

The childs are arranged according to the heuristic value H(n) so that the search starts with the one who is most likely to have the given keywords.The particular application and the properties of the encrypted files and keywords will determine how the heuristic functions f(n) and g(n) and the weighting factors $\alpha and \beta$ are actually implemented.
Rank(S,Q)=to w1 * OR(Synonym1,Synonym2,..., SynonymN) + w2 * AND(Keyword1, Keyword2,..., KeywordN) - w3 * NOT(UnwantedKeyword1, UnwantedKeyword2,..., UnwantedKeywordM)
Where, **Rank(S, Q)** is the ranking function for a search result S given a query Q.
w1, w2, and w3 are weights that determine the importance of each component in the ranking algorithm.
OR(Synonym1, Synonym2, ..., SynonymN) is the OR operation applied to synonyms of the keywords.
AND(Keyword1, Keyword2, ..., KeywordN) is the AND operation applied to the keywords.
NOT(UnwantedKeyword1, UnwantedKeyword2, ..., UnwantedKeywordM) is the NOT operation applied to exclude unwanted keywords.

B. Ranking Mechanism

A relevance scoring technique is used to organise the search results, taking into account several aspects including keyword frequency, synonym inclusion, and heuristic ratings derived from the DFS traversal[23]. This procedure ensures that the texts that are most relevant are retrieved first.

Greedy DFS with Ranking Algorithm: Input: rootNode, keywords, synonyms, alpha, beta.
Output: Ranked list of relevant nodes (files)
Result: a prioritised set of relevant nodes (files)
step1:create a blank list for rankedFiles.
Step 2: Use keywords, synonyms, alpha, and beta to define H(n).
Call GreedyDFSVisit(rootNode) in
Step 3 and follow the GreedyDFSVisit(node) procedure.
Next step 4: in case node is a file:
Step 5: Use H(node) to calculate the relevance score.
Step 6: Enable rankedFiles with a node and score
Step 7: Alternatively: Children = GetEncryptedChildren(node) is the eighth step.
Step 9: Arrange kids according to H, then in decreasing order
Step 10: for the kid inside the child:
Step 11: Should the youngster not get a visit:
Step12: Record the child's visitation
Step13: GreedyDFSVisit(child)
Step 14: Sort the ranked files in decreasing order of relevance score.
Step 15: Provide the ranking files back.

7. RESULTS AND DISCUSSION

The performance was evaluated using the REUTERS-21578 dataset, with parameters as detailed in this table 2 outlines the parameter settings for the analysis[24].
The table reference on next page. The k-means technique, which was implemented in Python, was used to cluster the texts into five or ten groups. Table 2 displays the number of texts for each cluster. Between 1,000 and 10,000 texts were used for the comparative study. A 2688-bit binary index was created by concatenating the results of several hash algorithms; this was lowered to 448 bits by using a reduction factor of 6. As originally described by Orencik and Savas [33,42], the proposed approach provides effective conjunctive searching using keyword field-free indexes. It is different from the current method in that it looks at fewer texts to get relevant results; instead, it looks at texts that are part of the matching cluster, which cuts down on comparison counts and search time. The current system, which was implemented with the identical parameters as listed in Table, was compared to the suggested scheme [33, 42].

A. Search Efficiency

To fetch the texts the users share the r-bit long query with CS. The texts are distributed into clusters leading to two possibilities regarding the occurrence of the texts in the cluster: Hard clustering: In hard clustering, a text appears only in one cluster.

A search scheme must offer high accuracy and efficiency to be considered for practical use.[25] The proposed search scheme improves search efficiency by reducing the average search time required to find relevant texts, unlike the existing schemes [33,42] which necessitate scanning the entire text collection. To evaluate search accuracy, metrics like recall, precision, F1 score, and False Accept Rate (FAR) were computed. We performed a test using 100 queries, each containing 5 relevant and 30 unrelated terms, on the text collection." In the figure 7 to represent the Search Accuracy Comparison Let's create a revised table to clearly present the search accuracy data based on the information provided.

The table 3 and 4 show the search accuracy comparison between the proposed scheme and the existing ones, and tools and technology indicating improvements in precision, F1 score, and a reduction in the False Accept Rate. The recall remains the same for both schemes at 100%.The gain column represents the percentage increase or decrease

TABLE II. Simulation environments with parameter

Dataset Name	Cluster Count	Number of Texts	Hash Function for Indexing	HMAC Functions for Query Construction	Reduction Factor (d)	Final Query Length (r)	Server Configu-ration)	Programming Language
REUTERS-21578 dataset [320]	JUniform: 5, Non-uniform: 10	1,000 to 10,000	MD5	SHA-256, SHA-384, SHA-512	6	448 bits	Intel Xeon Processor, 4 TB Hard Drive, 64 GB RAM	Python

TABLE III. Comparative analysis proposed Scheme and existing scheme

Parameter	Proposed Scheme	Existing Scheme	Gain
Recall	100%	100%	Same
Precision	82.4%	76.27%	+6.13%
F1 Score	89.07%	84.89%	+4.18%
FAR	0.128%	0.286%	-55.24%

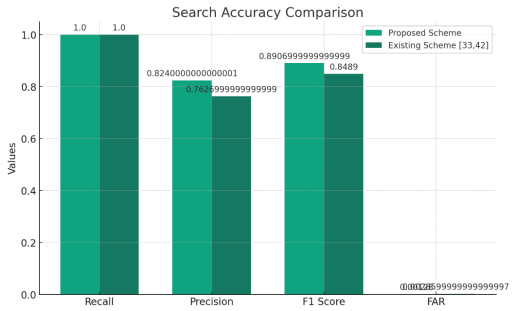

Figure 4. Search Accuracy Comparison

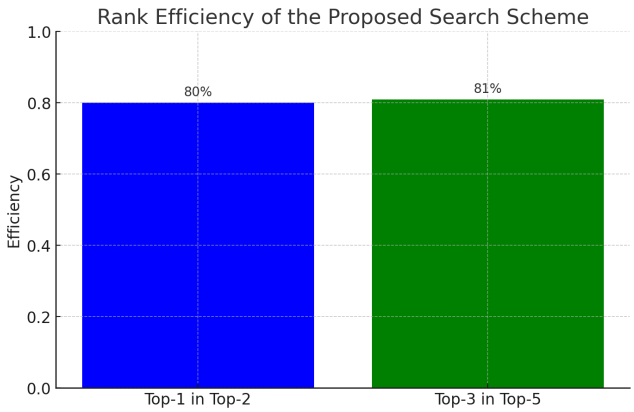

Figure 5. rank efficiency of proposed search scheme

in the performance of the proposed scheme compared to the existing schemes. [26]Our intended course of action involves assessing the efficacy of our suggested methodology by employing a cloud-based simulation environment. This evaluation will be conducted utilising a diverse range of datasets that encompass encrypted texts. The evaluation of the search's effectiveness will be conducted based on precision, recall, and computational time. [27]In order to showcase the enhancements in search relevance and efficiency, we will conduct a comparative analysis of our technique with the currently existent search schemes.

B. RANK EFFICIENCY

The efficiency of result ranking is evaluated by comparing the time needed to generate per-text 'p' indexes at various relevance levels within the text collection. [28]The increase in index build time associated with higher relevance levels is a one-time overhead, mitigated by the one-off nature of the indexing process conducted by the Data Owner (DO) during the offline stage. Cloud resources and parallel processing can be leveraged to further reduce this impact. Consequently, the extra time required for creating multiple indexes for each text is outweighed by the advantage of delivering superior ranked search results to the users.[29] To visualize this concept, we can create a graph that demonstrates the efficiency of ranked search results without specifying exact values. Let's plot a graph showing the proportion of top-ranked texts from the proposed scheme that align with the top results from plain-text searches. We'll use hypothetical data to illustrate the concept described.

8. CONCLUSION

The research conducted on the advancement in the field of secure recovery of data from cloud platforms. The proposed methodology integrates the resilience of greedy depth-first search algorithms with a sophisticated synonym identification system in order to offer accurate and efficient search functionalities across encrypted datasets. The method being described effectively addresses the challenges posed

TABLE IV. Tools and Technology

Tool/Technology	Purpose	Description
Encryption Software	Data Security	Software used to encrypt the cloud data. e.g.AES, RSA,
Cloud Platform	Data Hosting	Cloud service provider used to host the encrypted data e.g.AWS, Azure, Google Cloud, etc.
Indexing Engine	Data Retrieval	Tool used to create searchable indexes for the encrypted data.

by synonymy and polysemy in search queries, thereby guaranteeing users access to comprehensive results that are not only pertinent to the specific terms employed but also to their semantic counterparts. The significance of this matter is particularly pronounced inside the realm of encrypted data, as conventional search methods are inadequate in light of the limits imposed by privacy preservation. the incorporation of a rating system inside the search process facilitates users in efficiently identifying the most relevant texts, hence augmenting the usability of cloud storage services. By implementing encryption techniques to handle privacy issues, while also ensuring a high degree of search accuracy and efficiency, the proposed method effectively fills a significant void in the utilisation of cloud data. The algorithm's efficacy, as evidenced by many performance measures, underscores its potential for extensive implementation in secure cloud-based applications. Given the escalating prevalence of cloud services, the concurrent rise in data privacy issues necessitates timely and crucial study to safeguard data security and accessibility in the future. Future research endeavours may further enhance this groundwork by delving into machine learning algorithms to achieve more refined synonym detection. Additionally, adaptive ranking techniques based on user feedback might be explored to optimise the system's performance. Furthermore, the scalability of the system should be investigated in light of the escalating demands for cloud storage. The continuous endeavour to achieve perfection in the development of search systems that are secure, efficient, and intelligent poses a persistent challenge. This research serves as a significant advancement in this ongoing goal.

References

[1] M. A. H. D. A. Hosseingholizadeh, F. Rahmati and X. Liu, "Privacy-preserving joint data and function homomorphic encryption for cloud software services," *IEEE Internet of Things Journal*, pp. 728–741, 2024.

[2] Z. Fu, X. Sun, N. Linge, and L. Zhou, "Achieving effective cloud search services: multi-keyword ranked search over encrypted cloud data supporting synonym query," *IEEE Transactions on Consumer Electronics*, vol. 60, no. 1, pp. 164–172, 2014.

[3] C. Huang, D. Liu, A. Yang, R. Lu, and X. Shen, "Multi-client secure and efficient dpf-based keyword search for cloud storage," *IEEE Transactions on Dependable and Secure Computing*, 2023.

[4] Y. Z. H. H. M. Song, Z. Hua and X. Jia, "Lsdedup: Layered secure deduplication for cloud storage," *IEEE Access*, vol. 73, pp. 422–435, 2020.

[5] N. S. Joshi, K. P. Sambrekar, J. Abhijeet, S. Allagi, U. Patil *et al.*, "A hybrid approach for improving data security in cloud computing using greedy dfs ranked searching," *International Journal of Intelligent Systems and Applications in Engineering*, vol. 11, no. 9s, pp. 708–717, 2023.

[6] H. Cui and X. Yi, "Secure internet of things in cloud computing via puncturable attribute-based encryption with user revocation," *IEEE Internet of Things Journal*, 2023.

[7] J. W. Y. G. J. F. N. Wang, W. Zhou and J. Liu, "Secure and efficient similarity retrieval in cloud computing based on homomorphic encryption," *IEEE Transactions on Information Forensics and Security*, 2023.

[8] B. M. Nguyen, T. Nguyen, Q.-H. Vu, H. H. Tran, H. Vo, H. T. T. Binh, S. Yu, Z. Wu *et al.*, "A novel nature-inspired algorithm for optimal task scheduling in fog-cloud blockchain system," *IEEE Internet of Things Journal*, 2023.

[9] J. N. M. L. X. Zhou, D. He and X. Huang, "Single-server public-key authenticated encryption with keyword search and its application in iiot," *IEEE Transactions on Network Science and Engineering*, 2024.

[10] Y. Zhang, C. Jiang, and P. Zhang, "Security-aware resource allocation scheme based on drl in cloud-edge-terminal cooperative vehicular network," *IEEE Internet of Things Journal*, 2023.

[11] Y. Miao, Y. Yang, X. Li, L. Wei, Z. Liu, and R. H. Deng, "Efficient privacy-preserving spatial data query in cloud computing," *IEEE Transactions on Knowledge and Data Engineering*, 2023.

[12] C. Chi, Z. Yin, Y. Liu, and S. Chai, "A trusted cloud-edge decision architecture based on blockchain and mlp for aiot," *IEEE Internet of Things Journal*, 2023.

[13] Z. Xia, Q. Gu, W. Zhou, L. Xiong, J. Weng, and N. Xiong, "Str: Secure computation on additive shares using the share-transform-reveal strategy," *IEEE Transactions on Computers*, 2021.

[14] W. Dai, J. Liu, Y. Zhou, K.-K. R. Choo, X. Xie, D. Zou, and H. Jin, "Prbfpt: A practical redactable blockchain framework with a public trapdoor," *IEEE Transactions on Information Forensics and Security*, 2024.

[15] J. Han, L. Qi, and J. Zhuang, "Vector sum range decision for verifiable multi-user fuzzy keyword search in cloud-assisted iot," *IEEE Internet of Things Journal*, 2023.

[16] Z. Yang, B. Xiong, K. Chen, L. T. Yang, X. Deng, C. Zhu, and Y. He, "Differentially private federated tensor completion for cloud-edge collaborative aiot data prediction," *IEEE Internet of Things Journal*, 2023.

[17] D. Das, R. Amin, and S. Kalra, "Algorithm for multi keyword search over encrypted data in cloud environment," in *2020 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2020, pp. 733–739.

[18] H. He, J. Liu, J. Gu, and F. Gao, "An efficient multi-keyword search scheme over encrypted data in multi-cloud environment," in *2022 IEEE 7th International Conference on Smart Cloud (SmartCloud)*. IEEE, 2022, pp. 59–67.

[19] J. Li, J. Ma, Y. Miao, R. Yang, X. Liu, and K.-K. R. Choo, "Practical multi-keyword ranked search with access control over encrypted cloud data," *IEEE Transactions on Cloud Computing*, vol. 10, no. 3, pp. 2005–2019, 2020.

[20] B. Lang, J. Wang, M. Li, and Y. Liu, "Semantic-based compound keyword search over encrypted cloud data," *IEEE Transactions on Services Computing*, vol. 14, no. 3, pp. 850–863, 2018.

[21] P. Balamurugan, G. Arulkumaran, S. Jayagopalan *et al.*, "Multi-keyword graded exploration in encrypted cloud data for industries based on rc4+ and forest," in *2023 1st International Conference on Innovations in High Speed Communication and Signal Processing (IHCSP)*. IEEE, 2023, pp. 531–535.

[22] X. Yang, G. Chen, M. Wang, T. Li, and C. Wang, "Multi-keyword certificateless searchable public key authenticated encryption scheme based on blockchain," *IEEE Access*, vol. 8, pp. 158 765–158 777, 2020.

[23] Z. Xia, X. Wang, X. Sun, and Q. Wang, "A secure and dynamic multi-keyword ranked search scheme over encrypted cloud data," *IEEE transactions on parallel and distributed systems*, vol. 27, no. 2, pp. 340–352, 2015.

[24] L. Chen, Z. Chen, K.-K. R. Choo, C.-C. Chang, and H.-M. Sun, "Memory leakage-resilient dynamic and verifiable multi-keyword ranked search on encrypted smart body sensor network data," *IEEE Sensors Journal*, vol. 19, no. 19, pp. 8468–8478, 2018.

[25] S. Prakash, N. Andola, and S. Venkatesan, "Secure access of multiple keywords over encrypted data in cloud environment using ecc-pki and ecc elgamal," in *2017 International conference on Public Key Infrastructure and its Applications (PKIA)*. IEEE, 2017, pp. 49–56.

[26] P. Pandiaraja and P. Vijayakumar, "Efficient multi-keyword search over encrypted data in untrusted cloud environment," in *2017 Second International Conference on Recent Trends and Challenges in Computational Models (ICRTCCM)*. IEEE, 2017, pp. 251–256.

[27] Z. Fu, X. Wu, C. Guan, X. Sun, and K. Ren, "Toward efficient multi-keyword fuzzy search over encrypted outsourced data with accuracy improvement," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 12, pp. 2706–2716, 2016.

[28] Z. Fu, K. Ren, J. Shu, X. Sun, and F. Huang, "Enabling personalized search over encrypted outsourced data with efficiency improvement," *IEEE transactions on parallel and distributed systems*, vol. 27, no. 9, pp. 2546–2559, 2015.

[29] H. Yin, Z. Qin, J. Zhang, W. Li, L. Ou, Y. Hu, and K. Li, "Secure conjunctive multi-keyword search for multiple data owners in cloud computing," in *2016 IEEE 22nd International Conference on Parallel and Distributed Systems (ICPADS)*. IEEE, 2016, pp. 761–768.

Narendra Shyam Joshi is a Research scholar in the department of Computer Science and Engineering at KLS, Gogte Institute of Technology, Belagavi, and Karnataka.Affilated to Visvesvaraya Technological University. He has teaching Experience of 15+ years. . He has published 10+ papers in National and International Journals. He Has 3 book for Diploma students to his credit. Has delivered Technical talks on Current trends and Technologies in various Engineering / BCA/12th Science colleges. . He Attended and presented papers in 15+ conferences at National and International Level. Member of Professional Bodies: LMISTE. He can be contacted at email: joshinarendra50@gmail.com

Kuldeep P. Sambrekar is Professor in the department of Computer Science and Engineering at KLS, Gogte Institute of Technology, Belagavi. He recieved his Ph.D in Computer Science from Visvesvaraya Technological University in 2020. He has teaching Experience of 17+ years. He has filed Two patent to his credit. He has published 25+ papers in National and International Journals. Has 2 book chapters to his credit. He was invited as session chair to many IEEE conferences. He was invited as a Resource Person in 10+ workshops. He Attended and presented papers in 15+ conferences at National and International Level. Member of Professional Bodies: LMISTE, CSTA Reviewer for 3 international Journals. He can be contacted at email: kuldeep.git@gmail.com

Abhijit J.Patankar is working as an Associate Professor in Information Technology Department at D.Y Patil College of Engineering Akurdi Pune He is having 23 Years of Teaching and Research Experience. He is a Member of Board of Studies SPPU Pune. He has completed PhD from Computer Science and Engineering from VTU Belagavi. His areas of Research are Cloud Computing ,Data Science we Technology and AI.He has published number of research papers in reputed Journals .He has also worked as Reviewer for various different Reputed Journals. He can be contacted at email: abhijitpatankarmail@gmail.com

Prajakta Ajay Khadkikar is an accomplished Assistant Professor with a distinguished career spanning over 15 years in technical innovative teaching. She possesses a wealth of expertise in conducting impactful research in computer science, evidenced by numerous publications in reputable journals. She leads extensive involvement in leading and contributing to cutting-edge research initiatives, particularly in areas such as machine learning, computer vision, and data analysis. Proficient in a range of programming languages including Python, Java, R, and C++. She can be contacted at email: pakhadkikar@pict.edu

Archana JaganathJadhav is working as Assistant Professor, at the Dr. D Y Patil College of Engineering, Management and Research, India. Previously she has worked as Assistant Professor at Alard College of Engineering and management, India. She received her M.Tech CSE degree in 2014 from JNTU, India. She can be contacted at email: archana.jadhav@dypiemr.ac.in