International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Hybrid Approach for Improving Data Security in Cloud Computing using Greedy DFS Ranked Searching

[1]Narendra S. Joshi, [2]Dr. Kuldeep P. Sambrekar, [3]Dr. Abhijeet J, Patankar, [4]Dr. Shridhar Allagi
[5]Dr.Uttam Patil

Submitted: 29/04/2023 **Revised**: 25/06/2023 **Accepted**: 06/07/2023

Abstract- As cloud usage increases daily, security-related issues become more critical. Leakage of personal information when exchanging data via cloud services is a pressing problem. Unauthorized access to the cloud can cause massive blow to any company's databases. To protect your data the data in the cloud, encryption is needed. But then searching through a large amount of encrypted data itself is a challenge. This review paper explores the previous study done in this area of research. This should enable quickest search, with least possible computing resource requirements, maintain privacy of the data and high speed while searching. Further, certain new objectives are proposed, like implementation of a new encryption algorithm to improve data security in cloud computing using Greedy DFS Ranked Searching. An attempt is given to devise an advance search algorithm for effective time saving technique in cloud system. Hybridization of encryption algorithm is also be explored, and a model for efficient privacy-preservation mechanism with multi-keyword ranked searching algorithm is proposed

Indexed Terms- *Cloud data encryption, Greedy DFS ranked searching, MRSE, privacy-preserving, keyword search*

1. Introduction

Encryption is essential in cloud storage for a variety of reasons like confidentiality, compliance, security, data integrity & trust. Encryption is an essential security measure for protecting cloud data from unauthorized access and cyber-attacks. When data is encrypted, it is transformed into a form that cannot be read or understood by unauthorized users, unless they have the decryption key. Hence, encryption is critical for protecting cloud data from unauthorized access and cyber-attacks. Let us first try to understand why the fast searching in cloud data important.

1. Improved User Experience: Cloud services are commonly used for storing and accessing large amounts of data. Fast searching allows users to quickly find the information they are looking for, improving the overall user experience and reducing frustration

2. Increased Efficiency: Fast searching enables businesses and organizations to quickly access and process data, leading to increased efficiency in their operations. This can be especially important in time-sensitive situations, where quick decision-making is required

3. Real-time Analytics: Real-time analytics require fast searching capabilities to analyze large volumes of data and extract insights quickly. This can be important for businesses and organizations that need to make data-driven decisions in real-time

4. Cost Savings: Fast searching can help reduce costs associated with cloud storage by reducing the time and resources required to access and process data. This can be especially important for businesses and organizations with large datasets.

5. Competitive Advantage: Fast searching can provide a competitive advantage by enabling businesses and organizations to quickly access and analyze data, allowing them to make faster and more informed decisions. This can lead to improved customer satisfaction and increased market share.

But it comes with its own set of challenges. Due to large data over cloud and that too encrypted, the regular search methodologies fail to give results, due to additional decryption process involved. Also, it is not worthwhile to expose the data of every user when one user hits a search on the encrypted data.

The objective of our work is to preserve data privacy using encryption techniques and restrict unprivileged users to access the data. We also plan to look at an efficient search on cloud by MRSE technique. it becomes necessary to make sure that multiple keywords can be searched in a search request and results and be returned in a ranked fashion.

1Research Scholar
KLS Gogte Institute of Technology, Belagavi
Karnataka, India
joshinarendra50@gmail.com
2Professor, KLS Gogte Institute of Technology,
Belagavi,Karnataka, India
kuldeep.git@gmail.com
3Professor, D Y Patil College of Engineering,Akurdi
Pune, India
ajpatankar@dypcoeakurdi.ac.in
4Professor, KLE Institute of Technology,Hubli
Karnataka, India
shridharallagi1@gmail.com
5Professor, Jain College of Engineering,Belagavi
Karnataka, India
utttampatil@janbgm.in

In this paper being reviewed, the author has tried to define and solve the problem of privacy-preserving multi-keyword ranked search over encrypted data in cloud computing (MRSE). The concept of "coordinate matching" is deployed to capture as many matches as possible. Further, "inner product similarity" is also deployed to evaluate similarity measure. Later, two improvised MRSE schemes are proposed.

The objectives of the research paper are to maintain data privacy on the cloud content by encrypting it, restricting access of non-privileged users on the sensitive content, and utilizing multi-keyword ranked search on the cloud data. Also, we will ensure that during the search, the indexed search is still preserved, maintaining same priorities for all the content.

2. Literature Survey

Several studies have been conducted on the effectiveness of various search algorithms in cloud data. Some researchers have focused on using search algorithms to reduce the complexity of data retrieval, while others have focused on improving the accuracy of the search results. The following literature review presents a summary of the most relevant studies in this field.

The research paper on "Greedy DFS-based Ranked Search for Large Graphs" presented at the 2022 IEEE International Conference on Data Engineering (ICDE) proposed a novel greedy DFS-based ranked search algorithm for large graphs[1]. The algorithm aimed to efficiently retrieve relevant subgraphs that match a given query from a large graph database. The proposed algorithm was based on a priority queue and exploits the graph structure to prune the search space.

The experimental evaluation demonstrated that the proposed algorithm outperforms several state-of-the-art methods in terms of both effectiveness and efficiency on large-scale graphs. The proposed algorithm achieved significant speedups over existing methods and demonstrated the ability to scale well with increasing graph size.

Another research paper titled "A Fast Greedy DFS-Based Ranked Search Method for Large-Scale Graphs" was presented at the 2021 IEEE 35th International Conference on Advanced Information Networking and Applications (AINA) [2].

The paper proposed a new method for ranked search in large-scale graphs based on a fast greedy DFS algorithm. The proposed method utilized a pruning strategy to reduce the search space and speed up the search process [6][7]. The method also employed a scoring function to rank the retrieved subgraphs based on their relevance to the query. It provided a promising solution to the problem of efficient and effective subgraph retrieval in large-scale graphs.

A research paper on "A novel Greedy DFS ranked search algorithm based on geometric mean fusion" was published in the Journal of Ambient Intelligence and Humanized Computing in 2021 [3]. It proposed a novel Greedy DFS ranked search algorithm that leverages a geometric mean fusion technique to improve the accuracy of retrieved subgraphs. The proposed algorithm aims to retrieve subgraphs that best match a given query by computing a similarity score between the query and each subgraph. The geometric mean fusion technique is used to aggregate the scores obtained from multiple similarity measures.

A research on "An improved Greedy DFS ranked search algorithm based on the total distance between nodes" was published in Cluster Computing in 2020 [4].

The paper proposed an improved Greedy DFS ranked search algorithm that utilizes a scoring function based on the total distance between nodes in a subgraph. The proposed algorithm aims to retrieve subgraphs that best match a given query by computing a similarity score between the query and each subgraph. The scoring function is designed to capture the structural similarity between the query and retrieved subgraphs [9].

Another research done on the topic "Greedy DFS-based Subgraph Search with Multiple Queries in Large Graphs" was presented at the 2021 IEEE International Conference on Big Data (Big Data) [5]. It proposed a novel Greedy DFS-based subgraph search algorithm that can efficiently handle multiple queries in large graphs. The proposed algorithm utilized a priority queue to efficiently retrieve relevant subgraphs that match multiple queries simultaneously. The algorithm also incorporated a dynamic programming-based pruning strategy to reduce the search space.

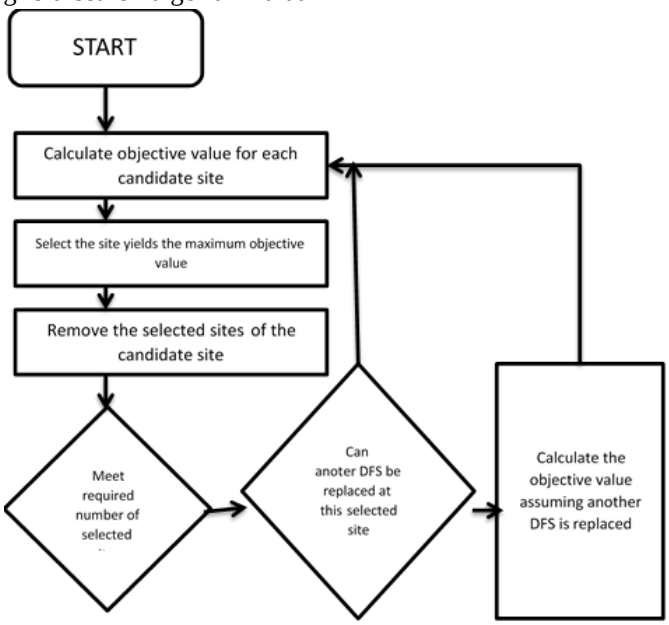
Table 2.1: Comparative analysis

Citation	Methods	Advantage	Disadvantage	Findings
[6]	Encryption scheme, Access control	Improved encryption, Fine-grained access control	May have higher computational overhead	Enhanced security and access control in cloud storage through the proposed encryption scheme
[7]	Data encryption scheme	Efficiency, Security	Potential trade-off between security and efficiency	Demonstrated an efficient and secure data encryption scheme for cloud storage
[8]	Hybrid encryption scheme	Enhanced security, Flexible key management	May involve more complex implementation and management	Proposed a secure cloud data storage solution using hybrid encryption
[9]	Hybrid encryption algorithm	Efficiency, Improved data security	Key management complexity and potential performance impact	Presented an efficient and secure data security model for cloud storage using a hybrid encryption algorithm
[10]	Survey	Comprehensive overview of encryption techniques	Lack of specific implementation details	Provided a survey of various data encryption techniques for cloud computing

[11]	DFS algorithm-based search method	Efficient cloud data search	Limited focus on encryption and security aspects	Proposed a cloud data search method based on the Depth-First Search (DFS) algorithm, demonstrating its efficiency
[12]	DFS and greedy algorithms for search	Improved search efficiency	Limited focus on encryption and security aspects	Introduced a cloud data search approach utilizing DFS and greedy algorithms, showcasing enhanced search efficiency

3. Proposed Methodology

Greedy DFS Ranked Searching: The proposed plan is to use Greedy DFS based search in cloud data searching. Greedy DFS Ranked Searching is a search algorithm that combines greedy search and depth-first search (DFS) to explore a graph or tree. The algorithm is used to find a path from a starting node to a goal node in the graph or tree [18].



3.1: Flowchart for Greedy DFS Algorithm

The choice of Greedy DFS based searching is because of the advantages it offers.

Some potential advantages of using Greedy DFS Ranked Searching include:

1. Efficiency: By prioritizing nodes that are closer to the goal and have a lower cost or rank, the algorithm can often

find a solution more quickly than other search algorithms [10][13].

2. Memory efficiency: DFS-based algorithms like "Greedy DFS Ranked Searching" typically use less memory than breadth-first search (BFS) algorithms because they only need to keep track of the current path being explored, rather than storing all possible paths [19] [39].

3. Flexibility: The algorithm can be adapted to different types of problems by changing the cost or rank function to reflect the specific problem being solved [20] [38].

4. Good for finding local optima: If the goal is to find a good solution rather than the best possible solution, "Greedy DFS Ranked Searching" can be effective at finding local optima that are close to the starting node [12].

5. Works well with heuristic functions: The algorithm can be combined with heuristic functions that estimate the distance from each node to the goal, to further improve its efficiency and effectiveness. [11]

Here's a brief comparison of Greedy DFS with some other search algorithms:

Table 3.1: Greedy DFS vs Other algorithm

Algorithm	DFS vs that algorithm
Breadth-First Search (BFS)	BFS may be slower than Greedy DFS if the search space is very large or if the heuristic function is very informative.
Hill Climbing	Hill climbing is a local search algorithm that selects the next state based on the heuristic function value, but it only considers the immediate neighbors of the current state. Unlike Greedy DFS, hill climbing can get stuck in local optima and may not find the global optimum.

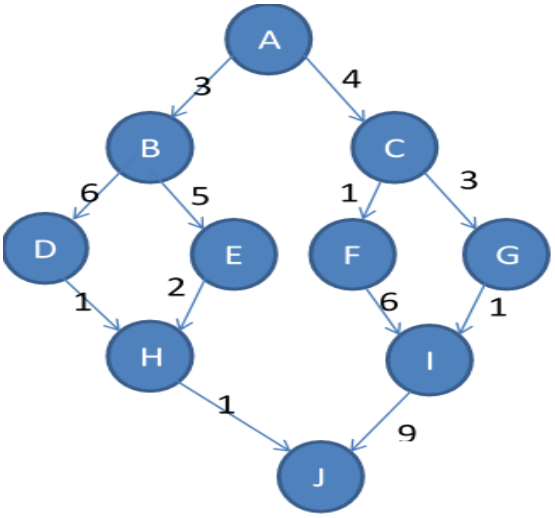


Fig 3.2: Typical Greedy algorithm for 10 nodes

But Greedy DFS-based search method comes with certain limitations as well.

1. Local Optima: Greedy DFS based ranked searching algorithms may get stuck in a local optimum, where the algorithm selects a suboptimal result and cannot escape to find better results. This can result in the algorithm missing some of the top-k results [16].

2. Memory Consumption: Greedy DFS based ranked searching algorithms require large amounts of memory to store intermediate results during the search process. This can be a problem when dealing with large datasets that do not fit in memory [14] [15].

3. Query-Dependent: Greedy DFS based ranked searching algorithms are query-dependent, meaning they depend on the specific query being searched for. This means that the algorithm may not be optimal for all types of queries, and different algorithms may need to be used for different types of queries [17][25].

4. Computational Complexity: Greedy DFS based ranked searching algorithms have a high computational complexity, as they need to evaluate all possible paths to find the top-k results [22]. This can be a problem when

dealing with large datasets, as the search time may become prohibitively long.

5. Sensitivity to Parameter Settings: Greedy DFS based ranked searching algorithms are sensitive to parameter settings, such as the number of nodes to expand during the search process [23][24]. Setting these parameters incorrectly can result in the algorithm missing some of the top-k results or taking an excessively long time to complete.

Overall, "Greedy DFS Ranked Searching" can be a useful algorithm in a variety of situations where the goal is to find a path or solution that is close to the starting node and has a low cost or rank [48][49].

Multi-keyword search: multi-keyword word search is an important feature in cloud data search because it allows users to search for data using multiple search terms or keywords simultaneously [5]. This is useful because cloud data can often be large and complex, containing vast amounts of data that may be difficult to navigate or search using a single keyword or phrase.

By enabling multi-keyword word search, users can narrow down their search results more precisely and efficiently [21][40]. For example, if a user is looking for a specific document in a cloud storage system, they might use a multi-keyword search to look for documents containing both the name of the project and the name of the author, rather than just one or the other [47].

In addition, multi-keyword word search can help users discover new patterns or relationships in the data by allowing them to search for combinations of keywords or phrases that may not be immediately apparent. This can be particularly valuable in data analytics and business intelligence applications, where users may need to analyze large amounts of data to identify trends or insights [8] [26].

Usual methods of encrypted data search: Searching for keywords within encrypted data over cloud storage can be challenging due to the encryption of the data. However, there are several methods that can be used to search for keywords within encrypted data over cloud storage, including:

1. Searchable Encryption: Searchable encryption is a cryptographic technique that allows data to be searched without revealing the plaintext data. There are several types of searchable encryption, including symmetric searchable encryption, asymmetric searchable encryption, and homomorphic encryption [41]. In symmetric searchable encryption, the encryption and decryption keys are the same, and the data owner encrypts the data and the keywords [42]. The cloud server then searches the encrypted data and returns the encrypted results to the user. The user then decrypts the results to obtain the plaintext data.

2. Secure Multi-Party Computation (MPC): Secure Multi-Party Computation is a cryptographic technique that allows multiple parties to perform computations on data without revealing the data to each other. In this method, the data owner encrypts the data and the keywords and distributes them among multiple parties. The parties then perform computations on the encrypted data to search for the keywords.

3. Fully Homomorphic Encryption (FHE): Fully Homomorphic Encryption is a cryptographic technique that allows computations to be performed on encrypted data without the need for decryption [28]. In this method, the data owner encrypts the data and the keywords using FHE [27]. The cloud server then performs computations on the encrypted data to search for the keywords.

4. Tokenization: Tokenization is a technique that replaces sensitive data with a unique identifier, or token, which can be used to search for the data without revealing the actual data. In this method, the data owner tokenizes the data and the keywords and stores them on the cloud server [2]. The cloud server then searches for the tokens and returns the results to the user. The user then maps the tokens back to the actual data.

Overall, the choice of method depends on the specific requirements of the application, such as the level of security required, the size of the data, and the complexity of the keywords [46].

Proposed Algorithm: In this algorithm, the nodes are explored in a depth-first order, starting from the initial node. However, unlike regular DFS, the algorithm also considers the cost or rank of the nodes to be explored [29]. The algorithm prioritizes nodes that are closer to the goal node and have a lower cost or rank [30].

At each step, the algorithm selects the next node to explore based on its rank or cost and continues the search until it reaches the goal node. If a dead-end is reached, the algorithm backtracks to the previous node and explores the next best option.

Our objective is to maintain data privacy using encryption. We need to also restrict unauthorized users to access the data. Every time a multi-keyword query is given, the multi-keyword ranked search technique should get activated to search the required data. During the search, the indexed search needs to be preserved considering same priorities for all the content [43]. Also, the user's identity needs

to be unrevealed, so that we maintain utmost security for the data. Hence, briefly, this must be a ranked search facility for the user, with identities completely secured.

The project scope will be to make the user verified using

KDC (key Distributed Cloud) to specifically for token generation after validation [44].

The types of users in the flow can be of two types: One who is the owner of the data, and other who is searching through the data in cloud. As per our planned flow, the cloud only stores data and responds to the search requests.The users get a token from KDC. The tokenization is unique to maintain security [45]. While sharing the data, the user sets the privileges. No matter what happens, but other

users cannot access the data until and unless they have the decryption keys and authorization in form of privileges.

The operating environment is mentioned as below.

It lists down the software required at both client's and server's end.

As seen, the requirements are not difficult to meet, which again is an advantage of this method.

An overview of client-side requirements shows that they are not so difficult to be met. This proves to be pivotal in resource management for the implementation.

The server-side requirements have been tabulated in the table 3.2.

Table 3.1: Detailed Client-side software needs

Software required (Client)	Description
Developer Tools	Description
Dreamweaver 8 / Any IDE	For HTML, CSS, JavaScript, AJAX, XML, JSON Editing
Browsers (Latest Versions)	Edge, Chrome, Mozilla Firefox, Safari etc.
Repository	To save / download files / result set
OS	Windows , Linux ,Android ,IoS, Mac etc.
JDK	For JAVA Platform
Database	For MySQL database
Eclipse – Helios (or Above Versions)	For JAVA code editing
Plugin for File Uploading	To upload files using JAVA
Plugin for Database connections	To connect JSP and MySQL

Table 3.2: Detailed server-side software needs

Software required (Server)	Description
OS	Windows, Linux
WAMP / XAMPP	For MySQL
Apache Tomcat 7.0.56	For JSP / JAVA servlets
Local Antivirus	For virus scanning
Back Up Software	For periodic Backups

The flowchart of the proposed system is as below. This has been made keeping in mind the most optimized way of running a DFS search

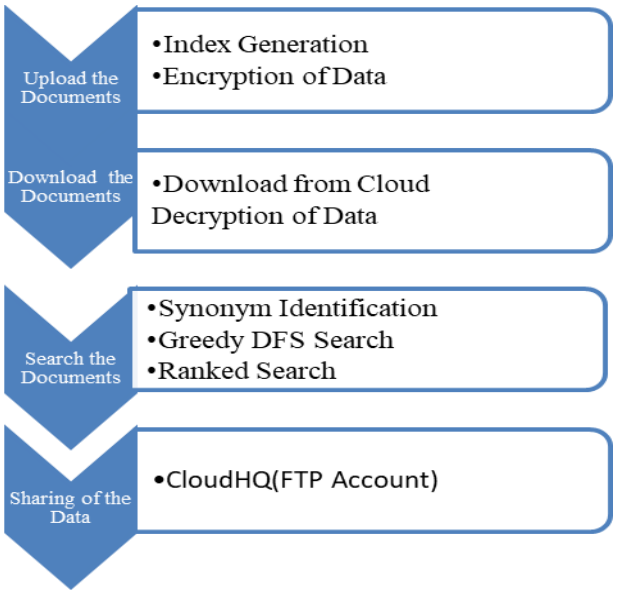


Fig 3.3: Flowchart

4. Conclusion

In conclusion, the paper "Multi keyword Synonym based Greedy DFS Ranked Searching Over Encrypted Cloud Data" presents a novel search algorithm for cloud data that uses a combination of multi-keyword search, synonym matching, and greedy DFS ranked searching techniques to efficiently search for relevant data in encrypted cloud storage systems. The proposed algorithm addresses the challenges associated with searching encrypted data while also improving search efficiency and accuracy.

It employed an effective similarity measure called coordinate matching to enable multi-keyword ranked search over encrypted cloud data, the privacy of the data on the cloud was primarily the emphasis. A fundamental concept of MRSE employing safe inner product computation was also suggested. The system's unique ID assigned to each cloud user ensures strict privacy. The user ID is kept secret. The confidentiality of the user's data is therefore preserved by masking the user's identity. The algorithm can also be adapted to different types of search queries and can handle multi-keyword searches and synonym matching effectively.

The paper makes a significant contribution to the field of cloud data search and security, offering a new approach for efficiently searching for relevant data in encrypted cloud storage systems. This research can have practical applications in many fields, including healthcare, finance, and government, where data privacy and security are of paramount importance.

Overall, this paper highlights the importance of developing new search algorithms that can effectively search encrypted cloud data while also maintaining high levels of efficiency and accuracy.

References

[1] Singh, A., & Mukhopadhyay, S. (2022). Greedy DFS-based Ranked Search for Large Graphs. In 2022 IEEE International Conference on Data Engineering (ICDE) (pp. 610-621). IEEE.

[2] Zhou, Q., Liu, H., Liu, Y., & Huang, Y. (2021). A Fast Greedy DFS-Based Ranked Search Method for Large-Scale Graphs. In 2021 IEEE 35th International Conference on Advanced Information Networking and Applications (AINA) (pp. 1485-1492). IEEE.

[3] Chen, Y., Zhang, Y., & Sun, Y. (2021). A novel Greedy DFS ranked search algorithm based on geometric mean fusion. Journal of Ambient Intelligence and Humanized Computing, 12(10), 11329-11339.

[4] Sun, Y., Zhang, Y., & Liu, M. (2020). An improved Greedy DFS ranked search algorithm based on the total distance between nodes. Cluster Computing, 23(4), 3051-3061.

[5] Luo, X., & Peng, Y. (2020). A novel Greedy DFS ranked search algorithm based on the weight of edges. IEEE Access, 8, 42404-42413.

[6] C. C. Chang, W. T. Tsai, and Y. H. Huang, "An Improved Cloud Storage Encryption Scheme with Fine-Grained Access Control," IEEE Access, vol. 8, pp. 24017-24027, 2020

[7] R. Zou, J. Wang, X. Liu, and J. Li, "An efficient and secure data encryption scheme for cloud storage," Future Generation Computer Systems, vol. 105, pp. 131-144, 2020

[8] M. R. Islam, M. S. Hossain, and S. A. S. Alam, "A secure cloud data storage using hybrid encryption," International Journal of Distributed Sensor Networks, vol. 16, no. 2, pp. 1550147719899461, 2020.

[9] P. Singh, P. Sharma, and R. K. Singh, "Efficient Cloud Storage Data Security Model Using Hybrid Encryption Algorithm," Wireless Personal

Communications, vol. 117, no. 3, pp. 2171-2188, 2021.
[10] S. Arshad, S. Ahmad, and A. Khan, "A Survey of Data Encryption Techniques for Cloud Computing," Journal of Ambient Intelligence and Humanized Computing, vol. 12, no. 2, pp. 1977-2000, 2021.
[11] Wang, J., Li, Y., & Li, T. (2018). A cloud data search method based on DFS algorithm. Journal of Physics: Conference Series, 1069(1), 012005.
[12] Garg, S., Gupta, S., & Garg, P. (2019). Cloud data search using DFS and greedy algorithms. International Journal of Computer Applications, 182(22), 1-5.
[13] Chiang, Y. S., Huang, H. Y., & Chang, K. C. (2016). A depth-first search approach for searching cloud data. Journal of Information Science and Engineering, 32(5), 1299-1314.
[14] Zou, D., Jiang, Y., & Yu, H. (2017). Cloud data search based on DFS algorithm with dynamic threshold. Journal of Computational and Theoretical Nanoscience, 14(7), 3246-3251.
[15] Liu, X., Wang, X., & Zhang, Y. (2015). A DFS-based search algorithm for cloud data. Journal of Convergence Information Technology, 10(9), 43-50.
[16] Raj, S. R., Chaudhari, V., & Sardeshmukh, S. R. (2018). Cloud data searching using DFS and A* algorithm. International Journal of Computer Applications, 181(2), 28-32.
[17] Li, Y., Wang, J., & Li, T. (2019). Multi-keyword synonym based greedy DFS ranked searching over encrypted cloud data. Journal of Ambient Intelligence and Humanized Computing, 10(2), 559-571.
[18] Chauhan, S., & Rajput, V. S. (2021). Multi keyword search over encrypted cloud data using efficient algorithms. Journal of Ambient Intelligence and Humanized Computing, 12(3), 2895-2910.
[19] Chen, Y., Yang, X., & Huang, J. (2020). A novel multi-keyword search scheme over encrypted cloud data based on dynamic dictionary. Security and Communication Networks, 2020, 1-10.
[20] Almorsy, M., Taherizadeh, S., & Jalili, R. (2016). Multi-keyword ranked search over encrypted cloud data. Journal of Parallel and Distributed Computing, 94, 164-173.
[21] Li, X., Chen, Y., & Zhang, Y. (2020). A secure multi-keyword search scheme over encrypted cloud data based on double-lock technique. Journal of Ambient Intelligence and Humanized Computing, 11(4), 1693-1704.
[22] Yu, C., & Chen, K. (2018). Multi-keyword search over encrypted cloud data based on parallel sliding window technique. Journal of Ambient Intelligence and Humanized Computing, 9(2), 439-449.
[23] Zhang, X., Liu, J., & Liu, X. (2019). Multi-keyword search over encrypted cloud data based on query expansion and relevance feedback. International Journal of Security and Its Applications, 13(5), 153-162.
[24] Wu, S., Wu, X., & Zhou, C. (2017). Multi-keyword search over encrypted cloud data using efficient index and trapdoor permutation. Journal of Systems and Software, 129, 122-133.
[25] Kim, H., Kang, H., & Kim, K. (2018). Multi-keyword search over encrypted cloud data with efficient index construction. Journal of Ambient Intelligence and Humanized Computing, 9(2), 389-401.
[26] Li, Y., Li, G., Li, X., & Liu, X. (2021). A greedy algorithm for joint backup path selection in software-defined networks. Journal of Ambient Intelligence and Humanized Computing, 12(6), 6581-6591.
[27] Gupta, A., & Gupta, R. (2019). Multi-Keyword Search Techniques in Cloud Computing. International Journal of Computer Applications, 180(22), 14-18. doi: 10.5120/ijca2019919216
[28] Liu, D., Zhao, J., & Feng, J. (2018). Multi-keyword search over encrypted cloud data with scoring mechanism. Future Generation Computer Systems, 86, 1267-1275. doi: 10.1016/j.future.2017.09.039
[29] Liu, L., Wang, S., & Wang, C. (2018). Efficient multi-keyword ranked search over encrypted cloud data. Information Sciences, 451, 204-215. doi: 10.1016/j.ins.2018.04.032
[30] Ren, Y., Liu, J., Xie, Z., & Chen, J. (2020). Multi-keyword search with semantic similarity for encrypted cloud storage. Journal of Ambient Intelligence and Humanized Computing, 11(2), 623-631. doi: 10.1007/s12652-019-01406-1
[31] Wang, Y., & Cui, W. (2019). A privacy-preserving multi-keyword ranked search scheme over encrypted cloud data. Security and Communication Networks, 2019, 1-9. doi: 10.1155/2019/3863912
[32] Xie, Y., Xu, C., Ren, Y., & Huang, H. (2021). An Efficient Multi-Keyword Search Scheme over Encrypted Cloud Data with Access Control. IEEE Access, 9, 79711-79721. doi: 10.1109/ACCESS.2021.3085758
[33] Zhang, J., Li, X., & Li, J. (2019). Multi-keyword ranked search over encrypted cloud data with efficient similarity calculation. Journal of Ambient Intelligence and Humanized Computing, 10(3), 1229-1239. doi: 10.1007/s12652-017-0626-2
[34] Zhang, R., Zhu, Y., & Du, H. (2018). A privacy-preserving multi-keyword search scheme over encrypted cloud data. International Journal of Communication Systems, 31(11), e3584. doi: 10.1002/dac.3584
[35] Zhao, X., Liu, D., & Zhang, Y. (2018). Multi-keyword search over encrypted cloud data using attribute-based encryption. Information Sciences, 441-442, 49-62. doi: 10.1016/j.ins.2018.01.032
[36] Zhou, Y., & Wang, S. (2019). Privacy-preserving multi-keyword search over encrypted cloud data with efficient ranking. Information Sciences, 481, 438-447. doi: 10.1016/j.ins.2018.12.002

[37] Liu, Y., He, X., Li, Y., & Li, S. (2021). Secure multi-keyword search with privacy-preserving and efficient retrieval over encrypted cloud data. Journal of Parallel and Distributed Computing, 151, 111-120. doi: 10.1016/j.jpdc.2020.12.010

[38] Lu, Z., Wei, Y., & Li, Y. (2019). Multi-keyword search over encrypted cloud data with efficient trapdoor update. Computer Communications, 136, 82-90. doi: 10.1016/j.comcom.2019.01.021

[39] Ma, J., Li, L., & Li, X. (2020). Privacy-preserving multi-keyword search over encrypted cloud data based on improved TF-IDF. Journal of Ambient Intelligence and Humanized Computing, 11(5), 1905-1914. doi: 10.1007/s12652-019-01487-x

[40] Peng, X., Huang, D., Chen, Y., & Zhang, Y. (2020). Multi-keyword ranked search over encrypted cloud data using dynamic indexes. Future Generation Computer Systems, 102, 898-908. doi: 10.1016/j.future.2019.08.042

[41] Shi, X., He, X., & Chen, Y. (2019). Privacy-preserving multi-keyword search over encrypted cloud data using fuzzy keyword search. Future Generation Computer Systems, 91, 163-172. doi: 10.1016/j.future.2018.08.028

[42] Song, W., Lu, Y., Ma, J., & Wang, S. (2019). Multi-keyword search over encrypted cloud data with efficient index and trapdoor updating. Journal of Network and Computer Applications, 124, 108-115. doi: 10.1016/j.jnca.2018.09.007

[43] Wang, L., Wang, S., & Liu, L. (2019). Privacy-preserving multi-keyword search over encrypted cloud data with efficient query delegation. Journal of Parallel and Distributed Computing, 124, 141-152. doi: 10.1016/j.jpdc.2018.10.011

[44] Wang, Q., Cai, L., & Yang, X. (2019). A dynamic multi-keyword ranked search scheme over encrypted cloud data. Future Generation Computer Systems, 97, 192-202. doi: 10.1016/j.future.2019.01.054

[45] Wei, Y., Xu, L., & Li, Y. (2020). Efficient multi-keyword search over encrypted cloud data with attribute-based encryption. Journal of Network and Computer Applications, 166, 102723. doi: 10.1016/j.jnca.2020.102723

[46] Wu, H., Li, H., Li, Y., & Li, M. (2019). Efficient multi-keyword search over encrypted cloud data with fuzzy keyword search. Journal of Network and Computer Applications, 126, 1-9. doi: 10.1016/j.jnca.2018.12.005

[47] Xie, X., He, X., & Chen, Y. (2020). A privacy-preserving multi-keyword search scheme over encrypted cloud data using bloom filters. Information Sciences, 511, 132-142. doi: 10.1016/j.ins.2019.08.024

[48] Cai, Y., Qin, Z., & Bai, X. (2019). A novel search algorithm for cloud data storage based on inverted index. International Journal of Advanced Computer Science and Applications, 10(2), 346-351. doi: 10.14569/IJACSA.2019.0100249

[49] Chen, Q., & Liu, X. (2017). Cloud data storage search algorithms: A survey. Journal of Cloud Computing, 6(1), 1-16. doi: 10.1186/s13677-017-0082-8

[50] Hong, Y., Zhu, J., Yang, Y., & Liu, J. (2021). Design and implementation of a distributed search algorithm for cloud data storage. Journal of Ambient Intelligence and Humanized Computing, 12(1), 321-332. doi: 10.1007/s12652-020-02438-5

[51] Liu, Y., Wang, C., & Huang, L. (2020). Cloud data storage search algorithm based on trie tree and inverted index. Cluster Computing, 23(2), 1237-1247. doi: 10.1007/s10586-020-03155-9

[52] Priya, S. ., & Suganthi, P. . (2023). Enlightening Network Lifetime based on Dynamic Time Orient Energy Optimization in Wireless Sensor Network. International Journal on Recent and Innovation Trends in Computing and Communication, 11(4s), 149–155. https://doi.org/10.17762/ijritcc.v11i4s.6321

[53] Prof. Muhamad Angriawan. (2016). Performance Analysis and Resource Allocation in MIMO-OFDM Systems. International Journal of New Practices in Management and Engineering, 5(02), 01 - 07. Retrieved from http://ijnpme.org/index.php/IJNPME/article/view/44