National College of Ireland

Enhancing Cloud Storage Security and Efficiency through Integrated Ranked Keyword Search and Cryptographic Techniques: A Multi-Client Approach

MSc Research Project
Cloud Computing

Vaishnavi Krishnananda Bhat
Student ID: x23110864

School of Computing
National College of Ireland

Supervisor: Yasantha Samarawickrama

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name: Vaishnavi Krishnananda Bhat

Student ID: X23110864

Programme: Cloud Computing **Year:** 2023-2024

Module: MSCCLOUD Research Project

Supervisor: Yasantha Samarawickrama

Submission Due Date: 12th August 2024

Project Title Enhancing Cloud Storage Security and Efficiency through Integrated Ranked Keyword Search and Cryptographic Techniques: A Multi-Client Approach

Word Count: 7665 **Page Count** 21

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Vaishnavi Krishnananda Bhat

Date: 12th August 2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Enhancing Cloud Storage Security and Efficiency through Integrated Ranked Keyword Search and Cryptographic Techniques: A Multi-Client Approach

Vaishnavi Krishnananda Bhat
x23110864

Abstract

Cloud storage services usage is growing to handle and store large amounts of data. This demands for advanced security measures that does not compromise data retrieval efficiency. Conventional encryption technologies even though secure, makes it difficult to perform effective searches, that results in trade-off between security and usability. This research work focuses on improving data retrieval relevance while maintaining strong security measures in a multi-client environment by integrating distributed point function (DPF) with frequency-based ranked keyword search. The system is implemented using Python and evaluated using a dataset that contains keywords of 4,000 documents. Key findings from this study shows that the frequency-based ranking algorithm greatly increases search relevancy while having minimal impact on overall efficiency. The system also maintains robust security by having low false positive and false negative rates. This work contributes to the development of more secure and efficient cloud storage solutions, addressing the growing needs of industries like healthcare and finance. Further research can focus on improvising on the scalability of the system and refinement of the ranking algorithms to better handle diverse and large-scale data environments.

***Keywords*— distributed point functions, ranking algorithm, data retrieval, multi-client, cloud storage**

1 Introduction

Cloud storage has increasingly become an essential component of modern computing, transforming the way the data is stored, accessed and managed. Conventional storage solutions depend on local hardware, whereas cloud storage enables users to store data on remote servers managed by third-party providers, which is accessible over the internet. Cloud storage technology provides numerous advantages over conventional storage systems due to which there has been growing use of services like Google Drive, Dropbox and AWS S3, which gives users the flexibility to access and share files from anywhere at anytime while also freeing up space on local devices.

According to Wasabi Technologies, Inc. (2024), 93% organizations plan to increase their public cloud storage capacity in 2024, an increase from 84% from the previous year. This is manly driven by growing data security needs, backup and recovery requirements

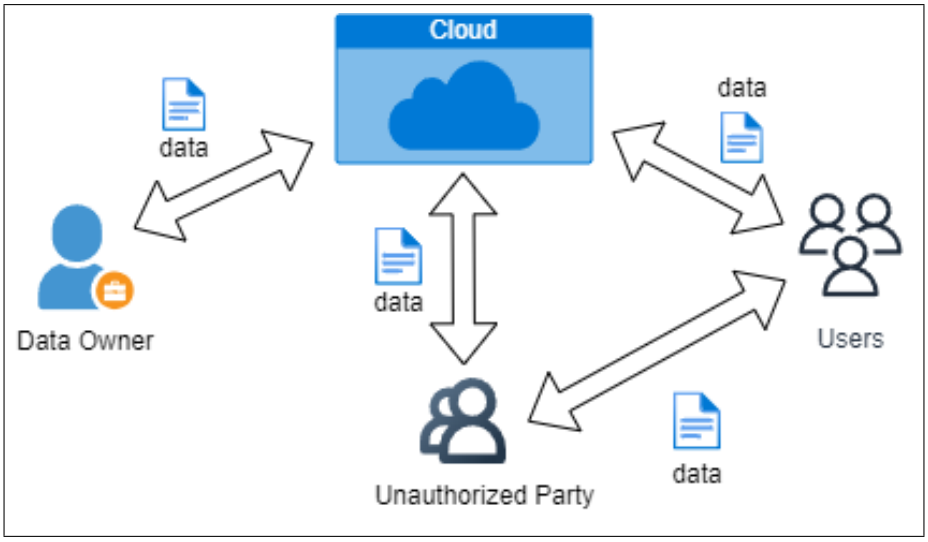

Figure 1: Cloud Sharing Environment

with Artificial intelligence (AI) and machine learning (ML) applications being a major factor. Its seen that 53% of businesses exceeded cloud storage budgets due to higher than expected usage and migration costs.

Features that make cloud storage so desirable, such as remote accessibility and large-scale data handling, also makes it vulnerable to security threats Yang et al. (2020). Many organizations store highly sensitive and confidential data in the cloud, such as customer records, intellectual property and financial information as shown in Figure 1. Tampering with such data results in severe consequences, ranging from financial losses to damage of organisation reputation Yang et al. (2020). Also, regulatory compliance bodies strictly require organizations to implement robust security measures to protect sensitive data on the cloud. Industries such as healthcare, finance and government are required to follow strict data protection regulations like the Health Insurance Portability and Accountability Act (HIPAA)[1] and Federal Risk and Authorization Management Program (FedRAMP)[2]. General Data Protection Regulation (GDPR)[3] is applied across the European Union, including Ireland, which applies strict requirements on how organizations collect, store, and process personal data Khalil et al. (2023).

Cloud data confidentiality and integrity is been threatened by cyberattacks, unauthorised access, and data breaches. Strong security measures is required for the storage of sensitive data, to protect against attacks, data breaches and unauthorised access Kumar et al. (2020) Kornaropoulos et al. (2020), while also maintaining the performance and functionality of the Cloud storage. Advanced encryption methods, fine grained access control and continuous monitoring of threats can be adopted ensure that security enhancements don't get in the way of the easy access and features that make cloud storage appealing. To make the most of cloud storage, it is important to balance security and efficiency, along with its functionality. End-to-end encryption makes sure that data is encrypted using users' cryptographic keys before being outsourced to cloud servers, ensuring that cloud servers store only ciphertexts. This method successfully protects the data from unauthorized access, even if attackers manage to compromise the cloud servers,

[1] https://www.hhs.gov/hipaa/index.html
[2] https://www.fedramp.gov/
[3] https://gdpr-info.eu/

they will get only encrypted information that is not readable without its corresponding decryption keys.

However, if straightforward cryptographic techniques are used, cloud servers will not be able to support keyword search, one of the essential features that plaintext storage providers offer. Zarezadeh et al. (2020) This research aims to address this challenge of effective keyword search by using advanced cryptographic techniques and enhance search functionality through ranked keyword search. This work is an attempt to improve search mechanism in cloud storage services, by building on the foundational work by Huang et al. (2023), that created a multi-client secure and efficient keyword search method for cloud storage using Distributed Point Function (DPF). Their work showed that safe keyword searches are feasible across multiple clients and along with ensuring that data remains secure even during search. This research focuses on building on top of their work with expressive keyword search, by integrating frequency based ranked search capabilities to enhance user experience by improving result relevance. The aim is to provide a scalable search solution that is user friendly and provides relevant search results in a multi-client environment in efficient manner. This study contributes to the system that complies with the regulations, protects user information and enables organizations to meet their legal obligations.

Solving this problem is important as it improves the usability of cloud services, which has a growing amount of sensitive data. Users need solutions that not only safeguard their information but also enable efficient, flexible access via advanced search features. In an effort to maintain the security and privacy of cloud data, Huang et al. (2023)'s solution concentrated on attribute-based access controls and multi-client keyword searches. This study offers an improvement on the work by Huang et al. (2023), by investigating DPF-based expressive keyword search—that is, ranked search. The novelty of this work is enhancement on their work with frequency based ranking and its integration with DPF-based keyword search. This aspect has not received much attention in previous works and proposed method addresses a significant gap in current systems by improving search relevance while also maintaining strict security protocols Gupta et al. (2022). The solution offers to balance trade-off between security, privacy, and usability in cloud-based keyword search systems.

1.1 Research Question

How can cloud storage services be optimized to balance efficiency, security, and usability by integrating cutting-edge cryptographic techniques with frequency-based ranked keyword search ?

This study explores existing works and address the gap found. The objective of this research is to specifically use DPF with a frequency-based ranked keyword search mechanism. Then assess the impact of this approach on search performance, accuracy, and security metrics and provide a scalable and practical solution for secure and efficient data management in cloud storage systems.

1.2 Structure of the Paper

In this paper, Section 1 provides an overview of the purpose of research including research question, motivation, aim, hypothesis, and contributions of the research. Section

2 reviews previous works that have conducted relevant studies on data security issues and effective keyword search strategies on cloud storage. In Section 3, the approach of the research methodology is discussed, details of the procedures and techniques used. Section 4 mentions the design specification, techniques and frameworks used. Section 5 includes the implementation details, tools and technology used. In section 6, detailed analysis of the results and findings based on performance metrics is discussed. Section 7 has Conclusion and Future work which discusses the effectiveness and limitations of the study as well as future directions for further research.

2 Related Work

In this section, various studies on cloud storage security is discused. The section is further divided into subsections namely Cloud Storage Security,Cryptographic Techniques in Cloud Storage, Cryptographic Techniques for Enhanced Security and Efficiency.

2.1 Cloud Storage Security

Cloud storage offers scalable and remotely accessible solutions, but it also presents serious security challenges, specially when handling sensitive data involving several clients.

A study by Yang et al. (2020)highlights a number of security issues in cloud storage, such as data confidentiality, availability of data and its integrity, fine-grained access control, secure data sharing, data leakage resistance, data deletion and privacy protection. The study proposes using encryption technologies to transform plaintext data into ciphertext, thereby reducing unauthorized access risks. It also discusses about regulating data with stringent access controls and authentication mechanisms. Cryptographic techniques ensures the integrity of data stored in the cloud and facilitates secure data sharing, along with preserving user privacy. While encryption enhances security, it complicates data management, especially in multi-client environments where it has challenging key distribution and its management.

Stefanov et al. (2018) introduced the Path ORAM protocol, which provides strong security guarantees by ensuring that access patterns are hidden. This protocol is extremely simple and efficient, making it practical for cloud environments. However, its implementation can be computationally expensive, which may not be suitable for all applications.

2.2 Cryptographic Techniques in Cloud Storage

Implementation of cryptographic techniques ensures data securit in cloud. These techniques include symmetric, asymmetric and hybrid encryption methods, each of which deals with specific efficiency issues and security requirements. In symmetric encryption, same key is used for encryption and decryption and is known for its speed and efficiency in encryption with large sets of data with minimal computational cost, but key distribution and administration are significant challenges. Yang et al. (2020) study discusses how symmetric encryption can be implemented efficiently but also underlines its limitations with key management. Asymmetric encryption, which uses a pair of keys, that is public and a private, is used for safe data transport. But it is not well suited for huge data quantities. It is slower and needs more processing power, but it solves key distribution issues. Hybrid encryption combines advantages of both, uses symmetric encryption for

data because of its efficiency and asymmetric encryption to secure the encryption keys, that provides a balanced approach to security and performance. The implementation of hybrid encryption systems is difficult since it requires integration of symmetric and asymmetric techniques.

Multi-user dynamic searchable symmetric encryption (SSE) with compromised players was investigated by Chamani et al. (2021). Their study focuses on collusion attacks and solves vulnerabilities in multi-user environments. While suggested solution methods increase security within multiple user systems, it often result in increased computing costs and complexity Chamani et al. (2021). Compressed encoding for homomorphically encrypted searches proposed by Choi et al. (2021) provides improved security by preventing information leaking during searches. Even though this increases privacy significantly, it has more processing overhead and complexity, which does not make it practical for large-scale deployments.

DORY is a distributed trust encrypted search system created by Dauterman et al. (2020) which makes use of distributed cryptography approaches to improve cloud security and trust. But, the deployment and maintenance is difficult due to the system's dependency on several reliable entities, specially in decentralised environments. Searchable symmetric encryption (SSE) by Gui et al. (2023) suggested enhancements to increase both security and efficiency of the system. But, due to their complexity, it is still difficult to incorporate these enhancements into the current systems.

To address these limitations, this research Boyle et al. (2016)uses a DPF-based system, which is based on function shares rather than conventional encryption keys. Since none of the parties may access the entire function, if one server is compromised, the risk is reduced. Since the client distributes the shares, secure key distribution paths between the servers are not necessary. Additionally, this improves privacy because more dispersed servers decrease the likelihood of all servers cooperating.

2.3 Cryptographic Techniques for Enhanced Security and Efficiency

Secure search work can be broadly classified into two main groups. (1) effective keyword search strategies; Sun et al. (2020) Miao et al. (2019). The reason for better performance of these methods because they are usually based on lightweight cryptosystems such as symmetric encryption, which minimise computational and communication overheads. Even though these methods are effective, they have security issues, as it allows attackers to identify whether frequently occurring searches are linked with identical keywords and locate those particular documents which are connected to these searches. Even though direct leakage of data is prevented, experienced attackers can still use these pattern leakages to recreate the document content and keywords, which poses a serious risk to privacy. (2) Leakage-free techniques by implementing expensive, complex cryptographic algorithms into practice Choi et al. (2021) Stefanov et al. (2018). High levels of privacy protection is offered by advance cryptographic algorithms like homomorphic encryption and Oblivious RAM (ORAM), it provides strong privacy protection against the previously mentioned pattern leakages. However, this improved security comes with high computational and implementation costs, which makes these methods less feasible for widespread practical use.

A basic cryptographic method known as Searchable encryption (SE) allows users to encrypt the data before uploading it to the cloud servers as well as search it without

exposing the query. There have been many studies that use Symmetric searchable encryption (SSE) to offer high computational and better transfer speed for search purposes, especially when it is combined with modern data structures Gui et al. (2023). Studies by Miao et al. (2019)Sun et al. (2020) suggested multi-client SE systems to improve data sharing. These multi-schemes are developed by using public-key cryptosystems due to the adaptability of managing key in public-key environments that allows multiwriter or multireader model with restricted access control.

The computational efficiency of multi-client keyword search algorithms that use symmetric keys that are suitable for models with one-writer or multi-reader is better compared to public-key enabled methods . In this context, Sun et al. (2020) suggested a multiclient non-interactive SSE strategy dependent on oblivious cross tag techniques (OXT), where data owner may remain offline during an authorized client execute search. In contrast, the studies on attacks with client server collusion by Wang & Papadopoulos (2021) and Chamani et al. (2021) states that exploited clients can collaborate with servers for retrieving details outside the allowed authorised usage.

These problems were solved by combining oblivious RAM and oblivious Map with traditional SSE techniques to create a multiclient collusion-resistant SSE system. While it is useful for some specific applications, most of this kinds of safe search algorithms are vulnerable to fraudulent attacks which make use of search and access pattern vulnerabilities. One of them is is attack by file injection, where an attacker can retrieve a customers query by looking at the clients access pattern, by injecting a file into a clients outsourced database.

Research on search pattern leaks shows that attackers can recognize customers keywords with a high probability even if they just have statistics regarding the database of the clients. To counter against such attacks, safe search solutions usually include three techniques: private information retrieval (PIR), oblivious RAM (ORAM) and differential privacy (DP). In MIPR, The customer search query requests are arbitrarily split into xor shared binary data indexes which is completely dependent on data indexes as well as replicated databases that are outsourced to different cloud servers Dauterman et al. (2020) . MPIR security is based on decentralised trust. Combining symmetric cryptosystems and MPIR in cloud storage makes it more generalized to enable safe keyword search. Due to high processing overheads and computational efficiency on clients as well as server, there is a limitation to use of MPIR-based keyword search algorithm.

DPF has similarities to MPIR technique that allows user to retrieve an item from multiple servers in possession of identical databases without exposing which item is being retrieved, provided servers do not collude Boyle et al. (2016). Since DPF has high computational efficiency with remarkably low communication overheads, it is more effective than the current MPIR. DPF enables keyword searches to achieve both efficiency and security in . By leveraging DPF techniques, this work aims to provide a solution for multi-client ranked keyword search in cloud storage environments that balances efficiency and security.

Author	Methodology	Research Domain	Achievements	Limitations	Differentiation
Chamani et al. (2021)	Dynamic SSE scheme for multi-user environments	Searchable Symmetric Encryption, Multi-User Environments	Enhanced security and efficiency for dynamic SSE environments	Scalability issues in large multi-user setups	Focus on dynamic SSE with corrupted participants
Gui et al. (2023)	Analysis of attack vulnerabilities in encrypted databases	Cryptanalysis of Encrypted Databases	Identified major security vulnerabilities under uniform attacks	Focus on theoretical vulnerabilities without direct solutions	Detailed focus on uniform attack vulnerabilities
Yang et al. (2020)	Critical evaluation and proposed improvements to SSE	Searchable Symmetric Encryption	Exposed limitations of existing SSE models and suggested improvements	Limited focus on real-world applications and adaptive query handling	Critique of existing SSE models and exploration of new algorithms
Boyle et al. (2016)	DPF-based keyword search with frequency-based ranking	Cloud Storage, Searchable Encryption	Enhanced search relevance with frequency-based ranking	Increased computational overhead in larger datasets	Efficient and Secure keyword search in DPF-based search systems
Gui et al. (2023)	Analysis of attack vulnerabilities in encrypted databases	Cryptanalysis of Encrypted Databases	Identified security vulnerabilities and provided basis for future research	Lack of practical solutions to identified vulnerabilities	Focus on uniform attack vulnerabilities in encrypted databases
Miao et al. (2019)	Attribute-Based Keyword Search over Encrypted Data	Attribute-Based Encryption	Improved keyword search capabilities in encrypted cloud data	Scalability and performance limitations in large-scale environments	Focus on multi-authority environments for secure keyword search
Stefanov et al. (2018)	Oblivious RAM Protocol	Secure Cloud Storage	Introduced a secure protocol for RAM operations in cloud environments	High computational costs in large-scale cloud environments	Introduced efficient ORAM protocol for cloud storage
Sun et al. (2020)	Non-Interactive Multi-Client Searchable Encryption	Searchable Encryption	Realization and implementation of non-interactive encryption for multiple clients	Scalability and efficiency concerns in practical implementation	Implemented non-interactive searchable encryption for multiple clients
Wang & Papadopoulos (2021), Chamani et al. (2021)	Design of a collusion-resistant searchable encryption scheme with optimal precision and security	Searchable Encryption, Multi-user Environments, Collusion Resistance	Achieved high levels of security and precision in multi-user searchable encryption while resisting collusion	Potential performance trade-offs due to the complexity of ensuring both optimal security and precision	Introduction of optimal precision in collusion-resistant environments, filling a gap in existing literature
Dauterman et al. (2020)	Comprehensive survey of data security and privacy protection techniques in cloud storage	Cloud Storage, Data Security, Privacy Protection	Provided a detailed analysis and classification of existing security measures and proposed new directions for future research	Survey-based research; lacks empirical implementation or validation of the proposed future research directions	Unique focus on the emerging challenges in IoT, smart cities, and digital transformation with respect to cloud storage security

Table 1: Summary of related works

3 Research Methodology

This section discusses the approach used and the process and techniques used to build the model. Model is designed to use the existing DPF framework and enhance it by adding ranking capbility to it so that it improve the relevance of search results, while also maintaining security and efficiency in a multi-client cloud environment. The three main stages of the research process was data collection and preprocessing, system implementation, and evaluation.

3.1 Research Procedure

DPF-based keyword search model was thoroughly studied, particularly the approach proposed by Huang et al. (2023), that uses garbled bloom filters and cuckoo hashing to achieve efficient and secure keyword searches over encrypted data.

Figure 2 depicts the research architecture and the techniques used. Ranking mechanism is directly integrated into the DPF framework. Integration involves modifying the search and retrieval process. Documents are ranked based on the raw frequency of the keyword within each document. The system architecture is modified to include a frequency based ranking that works on the server side. Frequency scores for each document that matches the search query is calculated and is integrated with encrypted keyword index and query handling process.

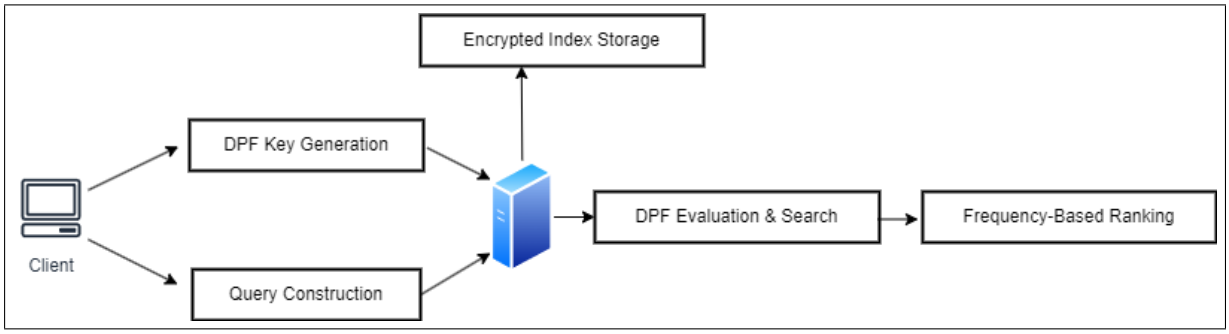

Figure 2: Architecture Diagram

3.2 System Implementation

Development and execution environment of the system is done using Python[4] in Google Colab [5]. Google Colab is chosen for its accessibility, ease of use and powerful computational resources which is useful for handling the intense tasks associated with cryptographic operations and data processing in the DPF system.

The first step in the implementation, was getting the data to be worked with, then figuring out the configuration required and the libraries to be installed. It involved importing necessary libraries, like Pandas for data manipulation and specific cryptographic libraries for handling encryption and decryption procedures. Python's extensive library support and ease of integration with different components is ideal for this project.

Integration of Google Colab with Google Drive made it easier to use the large dataset used in this system's evaluation. It is simple to upload, process and analyse dataset in the Colab, which makes it easier for execution of extended tests without requiring a special local setup.

- Data Collection and Preprocessing:
  Dataset used in this research is obtained from a publicly available dataset formatted in CSV file[6]. This dataset is a small subset of a real-world dataset from Wikipedia which contains 4,000 documents,each document has a maximum of 1,000 keywords

[4]https://www.python.org/
[5]`https://colab.google/`
[6]https://github.com/EnderCheng/KeywordSearch

extracted. Keywords were extracted from these documents using Python and KeyBERT[7] library.

- Ranked Search Mechanism:
  Search results are sorted by relevance, based on the frequency which allows users to receive more relevant search results and enhances the overall usability of the system. When many documents have the same keyword frequency, the current implementation will rank them based on the frequency. If they have the same frequency, they will appear together in the sorted list. Additional ranking criteria can be added such as document ID, timestamp, or relevance score.

- DPF Integration:

  DPF is a cryptographic primitive designed to enable two or more parties to combine and compute the evaluation of a point function without revealing the input point or the function value to any single party. A point function is a function that returns a specific value for one particular input and zero for all other inputs. DPF is defined for a function $f_{\alpha,\beta}$ where α is a specific input value (point) and β is the output value for that input. For all other inputs, the function returns 0.DPF then generates a set of keys (e.g., key0, key1) for each party Boyle et al. (2016). These keys are used to generate function shares that are distributed among the parties. Each party receives a share of the function, which by itself does not reveal any information about the function's specific input-output mapping. When a query is made, each party evaluates their function share using the provided key and the query input. The outputs from all parties are combined, typically using an XOR operation, to reconstruct the function's output. If the query input matches the specific input αdefined by the DPF, the combined output will be β. Otherwise, it will be 0. DPF allowes users to retrieve specific information from the cloud storage without revealing the exact search queries, thus maintaining search privacy.

  DPF is efficient in terms of communication and computational costs. Traditional Private Information Retrieval (PIR) protocols usually have substantial overhead, Whereas DPF significantly reduces this overhead. For insatnace, DPF-based schemes require fewer communication rounds and smaller key sizes compared to earlier PIR solutions. In the case of keyword search, It scales better with the size of the database. This is due to the efficient use of symmetric key operations, which are lightweight and can handle large datasets without significant performance degradation. This makes DPF particularly suitable for applications like secure keyword search in cloud storage. In Comparison with other multi-party computation or PIR techniques, DPF offers enhanced security against collusion attacks. DPF is constructed in a way that no single party or a collusion of some parties deduce the input or the output of the point function. DPF can be used for more complex applications such as range queries and generalized keyword searches, and this flexibility makes it an attractive choice for a wide range of privacy-preserving applications.

3.3 Evaluation Process

Efficiency and security are the two categories based on which implemented model is evaluated. Ranked search and unranked search system model is compared using various

[7] https://github.com/MaartenGr/KeyBERT

metrics. Under each category, multiple metrics evaluated are combined to get an assessment of security and performance of system. The results from this comparative study between the ranked and unranked search is used to determine the success of the system in meeting the objectives of the study.

- Efficiency Parameters:
  It is used to evaluate how quickly and efficiently the system can perform secure keyword search while having little computational overhead. In real time, it means how quickly the system can perform search of an index and give results. Metrics considered under this is DPF assessment time, search time, ranking time, indexing time and encryption time. Indexing and encryption times is a measure for system preprocessing efficiency, it helps in understanding how well the system will scale with bigger datasets.

- Security Parameters:
  It determines reliability and robustness of the model with protecting data integrity and getting accurate search results. False positive rate, false negative rate and encryption overhead are the metrics considered. False positive rate is used to understand the system's precision in identifying the presence of keywords within the documents. A lower false positive rate indicates a higher accuracy in search results and reduces the likelihood of irrelevant documents fetched. False negative rate measures the system's ability to retrieve all relevant documents. A higher false negative rate suggests that, the system is missing relevant documents. Encryption overhead is the additional computational burden added by securing the data, it is analyzed to ensure that the system's enhanced security features do not excessively hamper performance.

4 Design Specification

As represented in Figure 3, Three different entities make up a cloud-based data search system: clients, Cloud Servers (CS) and Data Owners (DO). Within the system, DO can create remote databases by outsourcing their documents to cloud servers. After the databases are created, clients approved by DO can query the documents linked to specific keywords, and cloud servers will provide a subset of documents that match the query rather than the complete databases.

In general, a remote document database can be split into two indexes Gui et al. (2023), as in Figure 4 and thus, the data search processes can be divided into two sections. Clients search a keyword index in the first section (Part-I) to obtain the identifiers of the documents that contain the keywords (highlighted with a red box). Clients get the content of the documents kept in the data index based on identifiers in the second section (Part-II). In this study, we primarily address Part-I, or keyword search, which is widely recognised as a "structure-only" keyword search model.

Under an outsourced model, clients are permitted to conduct unauthorised searches, i.e., their trust comes from authorization. Since semi-honest or hostile attackers may hack cloud servers, they cannot be completely trusted. The semi-honest CS will sincerely adhere to a safe keyword search strategy, but it might be curious and attempt to use passive attacks to retrieve DO's data and clients' enquiries. Additionally, the malicious CS may diverge from a safe search protocol in order to deceive the clients regarding the query

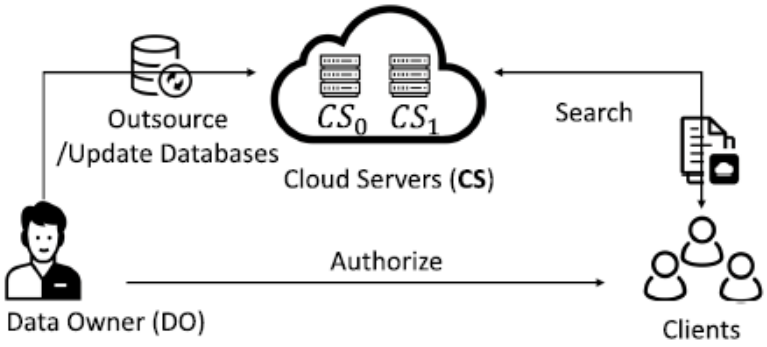

Figure 3: System model
Huang et al. (2023).

outcomes. Also, an adversarial cloud server collusion assault is taken into account, in which a portion of the clients work together to extract information beyond authorisation and hide cheating behaviour. Other malicious attacks on system availability, such as purposefully erasing client data and refusing client enquiries, are out of scope. In trust model. DO is always trusted, and making a reasonable assumption that at least one cloud service provider operates honestly and is not compromised by adversaries. Actually, DO can choose between two distinct cloud service providers to create this kind of environment. Consequently, this work considers a two-server model, represented by CS0 and CS1.

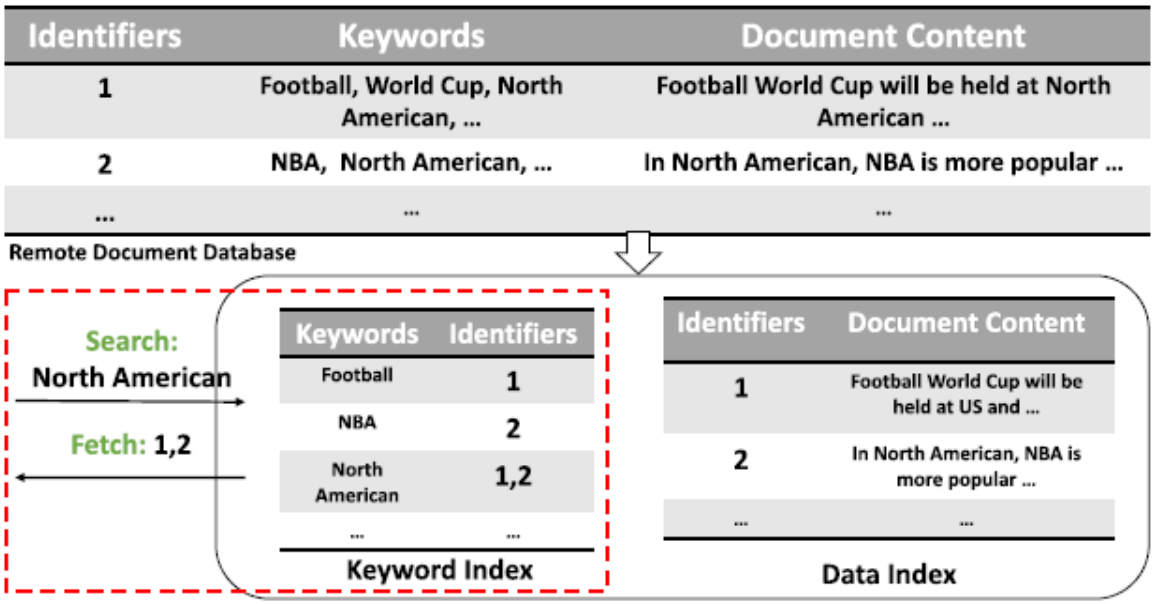

Figure 4: Outsourced data indexes Huang et al. (2023).

4.1 System Components

- DPF-Based Secure Keyword Search: The system uses DPF to allow clients to generate secure tokens for their search queries. These tokens are processed by the server without exposing the content of the search, preserving the privacy of the

client's query. The server then matches these tokens against an encrypted keyword index, identifying documents that contain the search terms without showing the data.

- Frequency-Based Ranking Algorithm: Once the relevant documents are identified with the DPF, they are ranked based on the frequency of the search terms. This ranking algorithm is simple yet effective, it provides a prioritized list of documents that are more likely relevant to the user's query. This ranking is done based on frequency of occurance where documents with higher occurrences of the queried keywords will be ranked higher. If multiple documents have the same keyword frequency, in the current implementation they will appear together in a sorted list. Additional ranking criteria can be added for more relevance such as document ID, timestamp or relevance score.

- Hashing: It is used to transform given data into a fixed-size string, SHA-256 is a cryptographic hash function that generates a fixed size 256-bit (32-byte) hash value from the input data. It is part of the SHA-2 family of hashing algorithms. It is used for creating encrypted indexes for secure keyword searches. It ensures that the mapping between keywords and document identifiers remains confidential. This prevents potential attackers from gaining insights into the contents of the cloud storage through the index, thereby maintaining the confidentiality of the data.

- Symmetric Encryption: It is provides data security by using the same key for both encrypting and decrypting data. In this work, it is used to encrypt document IDs during the search process.

4.2 System Architecture

It is built on a client-server architecture, in which the client creates and submits secure searches using DPF method and the cloud server handles search query processes and stores encrypted indexes.

- Client-Side Operations: Every client will generate a unique key to create DPF tokens for querying the encrypted index to make sure that the search queries are secure and private. Clients will produce search queries by selecting keywords and generating its corresponding DPF tokens, which will then be sent to the server for evaluation.

- Server-Side Operations: Server will maintain an encrypted index of documents, where every keyword is linked with a set of encrypted document identifiers. This is created in the indexing phase by processing documents to extract keywords, calculate their frequencies, and encrypt the information. On receiving a query, the server will evaluate the DPF tokens against the encrypted index, by identifying relevant documents without exposing the underlying data. After identifying the relevant documents, server will rank the results based on frequency of queried keywords in each document. This ranking is done server-side to optimize performance and reduce client-side computational load.

System is designed to meet specific security and performance requirements where all documents and keywords index are encrypted to prevent unauthorized access. DPF

ensures that the server cannot read the actual keywords or document contents from the search queries. Each client's queries are isolated from others to prevent collusion and information leakage. The system is developed to handle large datasets with fast indexing and search operations. The architecture is designed to support scaling to accommodate more documents and clients, maintain efficiency. The model processes large text datasets to extract keywords and calculate their frequencies, and ensure efficient and scalable index. Server securely manages encrypted indexes, ensuring they store and retrieve data without exposing. Model provides accurate and relevant search results, ranks documents based on keyword frequency and returns results promptly.

System processes the set of keywords of documents and calculates their frequency. Keywords and their associated document identifiers are stored in a secure index after encrypted using SHA-256 hashing. This encrypted data forms the basis for secure search operations and ensures that sensitive information is protected throughout the process. The implementation is done using Python for entire lifecycle of the keyword search process, from data loading, indexing, query processing and result ranking. The code has been constructed to be modular and extensible way, that will allow for easy adaptation to new datasets and cloud environments. The modularity ensures components can be independently tested and refined and also improves the overall robustness of the system.

5 Implementation

This section of the paper discusses the implementation that is carried out, the tools and technologies used. The main focus is on how frequency-based ranking algorithms and DPF can together provide a safe, efficient and reliable solution for keyword search. The implemented model processes a collection of keywords of documents and calculates their frequency. Keywords and its associated document identifiers are encrypted using SHA-256 hashing and stored in secure index. This data encryption ensures that sensitive information is protected throughout the process. Key functionalities are DPF token generation and evaluation, keyword frequency calculation, encrypted index management and ranking algorithm integration.Figure 5 depicts the Implemetation flow diagram.

Frequency-Based Ranking Algorithm:
The algorithm designed to rank search results based on the number of times a keyword appears in each document. System prioritizes documents where the keyword occurrance is more frequent and consider that those documents are more relevant to the user query.

- For each keyword, the system maintains a record of the documents it appears in and also how many times it appears in each document. This is done by creating keyword index, where each entry consists of a keyword, a list of document IDs where it appears and the corresponding frequency of the keyword in those documents.
- The system then calculates how many times a particular keyword appears for every document. This frequncy information is used for ranking.
- When keyword search is performed, the system gets all the documents containing that keyword. The documents then is ranked based on the frequency of the keyword. Documents with higher frequency of keyword appearance is given higher ranks.

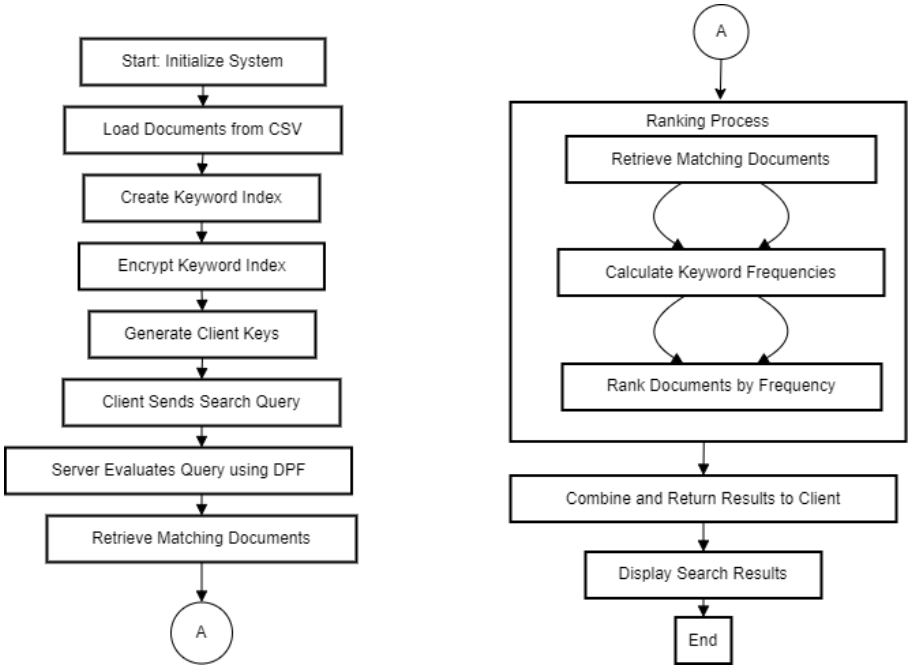

Figure 5: Implemetation flow diagram

- For privacy, keyword index and its associated frequency information is encrypted. This encryption ensures to protect content of the documents and the nature of the search.

5.1 Tools and Languages Used:

- Python: It was used create the complete system due to its extensive libraries and tools which makes data processing, algorithm development and cryptographic operations easier. Its chosen for its flexibility and ease of use made for implementing both the DPF-based search mechanism and the frequency-based ranking algorithm.
- Google Colab: It provides the required computational resources and has the ability to handle Python's extensive libraries. It streamlined the development process, allowing for efficient execution of the code in a cloud like environment.
- Hashlib: Python's hashlib library provided by SHA-256 is essential to the encryption of document identifiers and keywords. This cryptographic technique makes sure that every piece of data kept in the index is safely hashed to guard prevent illegal access.
- OS module: It is used for generating random cryptographic keys required for DPF. These keys are essential for secure encryption and reconstruction of document IDs during searches, that ensures that only authorized users can access sensitive data.
- Base64 encoding: It is used to safely encode the encrypted document IDs before storing or transmitting them. This encoding makes thee binary data remain intact and can be easily decoded when needed while also maintaining the security of the document identifiers across different systems and storage formats.

- JSON (JavaScript Object Notation) : It is used to store and manage performance metrics generated during the indexing and search processes. It stores metrics such as indexing time, encryption time and search performance in a structured JSON format. This the data can be easily analyzed, shared and compared. It helps to gain a deeper understanding of the system's efficiency and security.
- Libraries: Pandas is used for efficiently handling document collection. CSV library is used for reading and processing of input data files.

6 Evaluation

This section gives a detailed analysis of the experimental results, discuss the findings of the model implemented and evaluate their significance in terms of practical implications. The evaluation is mainly focused on metrics such as search efficiency, ranking accuracy and security. Table 2 shows the values obtained for all the metrics evaluated for keyword -medallists.

Metric	**Ranked Search**	**Unranked Search**
Search Time	0.009266	0.000789
DPF Evaluation Time	0.000646	0.000750
Ranking Time	0.000016	0
Indexing Time	40.937437	38.484231
Encryption Time	37.884078	36.841655
DPF Key Generation Time	19.861279	19.036669
False Positive Rate	0.363636	0.363636
False Negative Rate	0.125	0.125
Encryption Overhead	0.925414	0.957318
Keys Generated	1	1

Table 2: Comparison of Ranked and Unranked Search Metrics

6.1 Efficiency Analysis

For the keyword search - medallists, Figure 6 illustrates the efficiency metrics comparison. Efficiency of the model was measured first with DPF evaluation time and the additional time introduced by the ranking process. It was observed that the ranking slightly increases the total search time, but the overall impact is very minimal. Ranked search time is longer than unranked search due to additional ranking step involved in sorting the results. But, both times are extremely fast and ranking adds negligible difference for practical use. This means that the ranking process, which is designed to enhance the relevance of search results introduces negligible overhead, indicating that the system can be scaled effectively without compromising on the performance. Ranked search takes longer to index documents because of extra steps involved in calculating keyword frequencies, this adds some complexity. This also results in slightly higher encryption time. Overall, the slight increase in efficiency metrics is within the acceptable limits for real-time applications, which confirms that the model meets the performance expectations for enhanced secure keyword search in cloud environment.

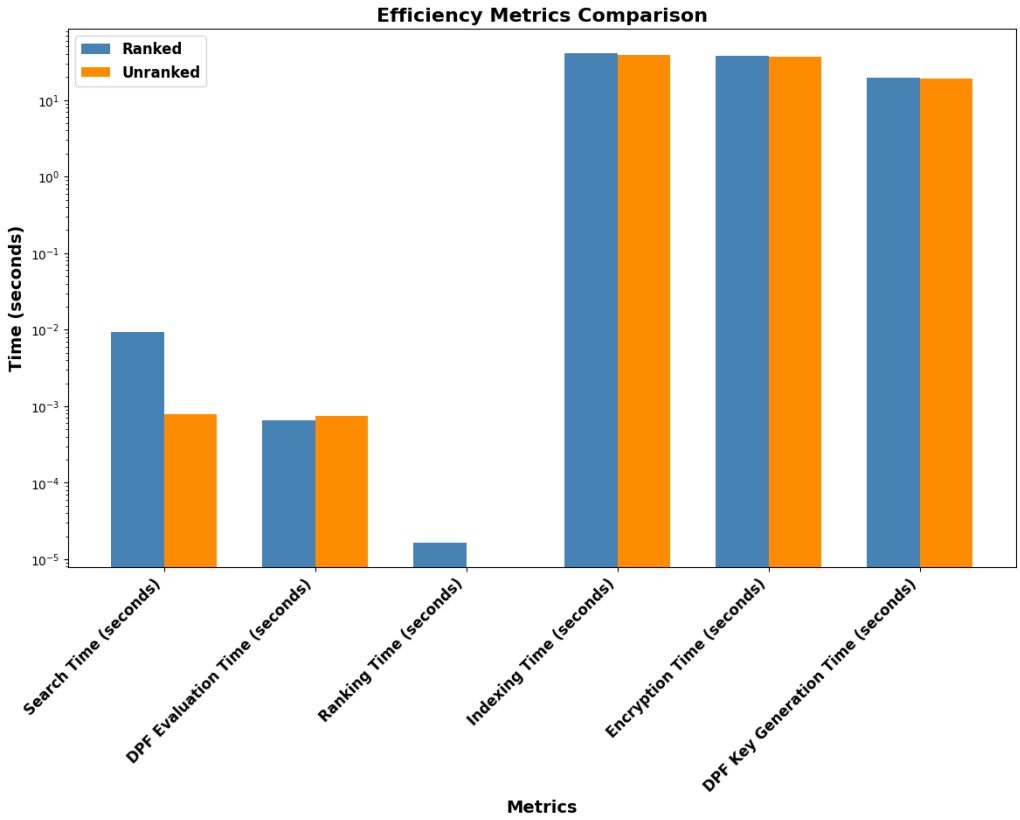

Figure 6: Efficiency Metrics Comparison

6.2 Security Analysis

A `true_positive_docs` list is set that contains the document IDs that are known to be relevant to the search query. This list is used as a reference to evaluate the accuracy of the search results. For calculating false positive and false negative rates for the keyword - medallists, Under true positive document, 7 actual true positive documents were set and 1 document that does not contain the keyword was set was set. Figure 7 illustrates the security metrics comparison for the keyword - medallists. Both the ranked and unranked searches have the same false positive rate, meaning that about 36% of the documents returned are not relevant to the search query. This shows that in terms of filtering out irrelevant results both approach performs the same. Even with false negative rate, it is the same for both showing 12.5% of the relevant documents. This shows that ranking process does not appear to affect the ability to retrieve relevant documents and maintains the same security. Encryption overhead was slightly higher in the ranked configuration but remained within manageable limits. It indicates that while the addition of ranking introduces some computational overhead, the system's strong security features do not deviate from its overall efficiency. The generation of encryption key is consistent across configurations and supports the system's ability to securely manage client queries without significantly degrading the performance.

Table 3 shows the comparison of ranked and unranked search metrics across different keyword with the same true positive set for medallists keyword. Hence, there is a drastic change in the false positive and false negative rates for the keyword - football and european, suggesting that the model works accurately.

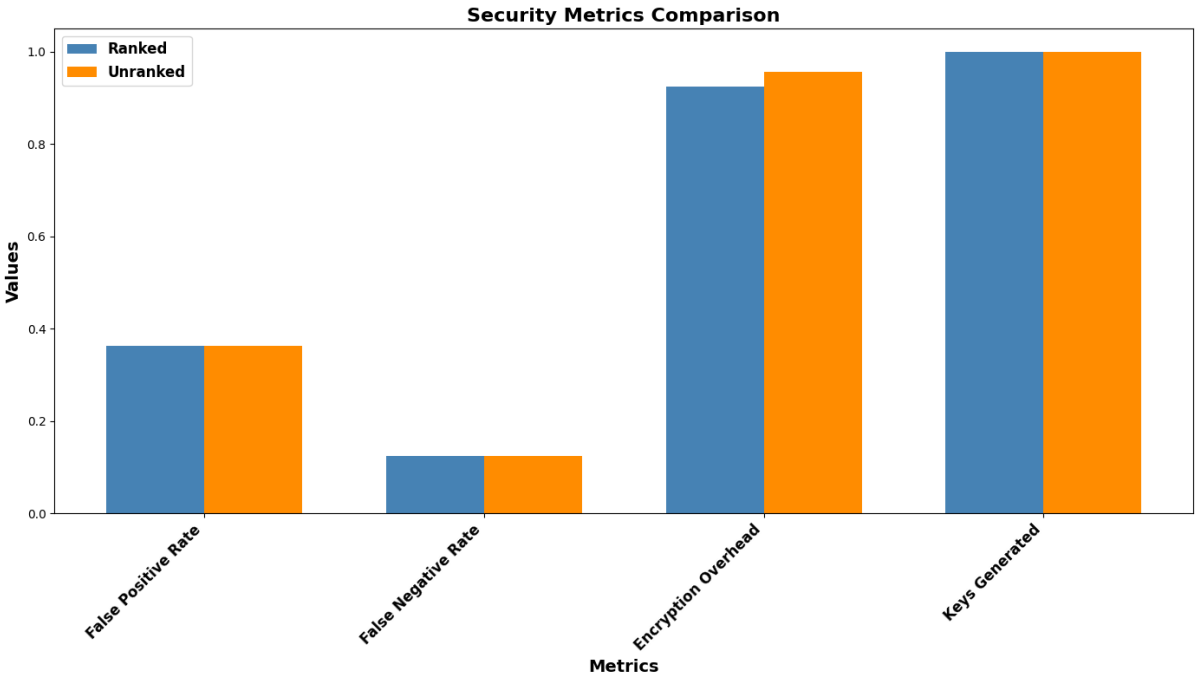

Figure 7: Security Metrics Comparison

6.3 Experiment 1: Keyword - medallists

Ranked search shows a significant increase in search time (0.009266 seconds seconds) compared to unranked search (0.000789 seconds). The DPF evaluation times are almost identical, indicating that the ranking process adds a small overhead. Indexing and encryption times were higher for ranked search, reflecting the additional processing needed. The higher search and evaluation times in ranked search reflect the additional computational effort required to provide ranked results, this comes with the benefit of more relevant search outcomes.

6.4 Experiment 2: Keyword - football

For keyword - football, ranked search drastically increased search time to 0.622294 seconds from 0.054644 seconds in unranked search, indicating a significant overhead for ranking. The DPF evaluation times were similarly impacted. Indexing and encryption times were also higher for ranked search. Ranked search maintained a false positive rate of 1, similar to unranked search, as the true positive that was set did not have any documents that contained the keyword. The encryption overhead was marginally lower, indicating efficient handling of the ranking. Ranked search improves accuracy but at the cost of significant computational overhead, especially for complex keywords like *football*. This trade-off must be considered in large-scale applications.

6.5 Experiment 3: Keyword - european

Ranked search for - european keyword resulted in a search time of 0.188779 seconds, which, while higher than the 0.029164 seconds for unranked search, remains very low and acceptable for practical use. The DPF evaluation and indexing times followed a similar trend, with slightly higher times for ranked search. Both ranked and unranked searches

had high false positive and negative rates, pointing to issues with accuracy. The encryption overhead was slightly lower for ranked search, but the impact is minimal. Ranked search's minimal impact on performance for simpler keywords like *european* suggests it's a viable option for such cases. However, the high error rates indicate a need for further optimization.

Keyword	**medallists**		**football**		**european**	
Metric	**Ranked Search**	**Unranked Search**	**Ranked Search**	**Unranked Search**	**Ranked Search**	**Unranked Search**
Search Time (seconds)	0.009266	0.000789	0.622294	0.054644	0.188779	0.029164
DPF Evaluation Time (seconds)	0.000646	0.000750	0.054081	0.054617	0.014641	0.029133
Ranking Time (seconds)	0.000016	0	0.000108	0	0.000054	0
Indexing Time (seconds)	40.937437	38.484231	43.813852	37.751134	39.289599	35.926143
Encryption Time (seconds)	37.884078	36.841655	39.825803	35.961712	36.572292	34.363321
DPF Key Generation Time (seconds)	19.861279	19.036669	20.228521	18.738825	18.971179	17.982066
False Positive Rate	0.363636	0.363636	1	1	0.975	0.975
False Negative Rate	0.125	0.125	1	1	0.375	0.375
Encryption Overhead	0.925414	0.957318	0.908977	0.9526	0.930839	0.956499
Keys Generated	1	1	1	1	1	1

Table 3: Comparison of Ranked and Unranked Search Metrics across different keywords.

6.6 Implications

Findings from this system evaluation have important implications for both theoretical research and practical implementation. From academic standpoint, the results contribute to ongoing research with respect to balancing security and performance in multi-client cloud storage systems. The integration of DPF with frequency-based ranking offers a novel approach to enhance search relevance while also maintaining security standards, providing a foundation for future research in secure data retrieval. The systems performance indicates that it can be effectively used in environments where data privacy is of utmost importance, such as healthcare,finance and legal services. The minimal impact of search performance on ranking, along with its strong security features, makes the system a desirable solution for safe and enhanced keyword search.

6.7 Discussion

DPF-based keyword search with integrated frequency-based ranking system emphasises a critical trade-off between security and efficiency. The system built successfully enhances the relevance of search results and maintains strong security, but this improvisation comes at the cost of increased computational overhead, mainly in search and DPF evaluation times. This trade-off is a common challenge in secure cloud storage systems, where adding

advanced cryptographic techniques and usability features often impacts performance. Even though this systems efficiency remains within acceptable limits, its scalability and response times may be affected as the dataset size and user base grows.

One of the limitation of this experiment is its dependency on a small dataset and controlled environment. The use of dataset containing 4,000 documents even though is enough for initial testing, it does not capture the full complexities and challenges that is seen in a real-world setting where datasets can be significantly larger and more varied. As the size of the dataset grows, the time required for indexing, encryption and search increases leading to scalability issues. Handling such large-scale data efficiently and maintaining security can be challenging. Distributed architecture where the dataset is partitioned across multiple servers can be used to manage large datasets. It enables parallel indexing and searching which can significantly reduce processing time. In addition, the experiment did not consider network latency, which can have a considerable impact on performance in cloud environments. Further work can be done with larger and more varied datasets and in environments that simulate real-world conditions more closely, that includes network delays and varies load conditions.

7 Conclusion and Future Work

The research question guiding this study was:**How can cloud storage services be optimized to balance efficiency, security, and usability by integrating cutting-edge cryptographic techniques with frequency-based ranked keyword search ?** . The objective of this research was to develop a secure keyword search system that enhances data retrieval efficiency and relevance while maintaining strong security measures. The system built was evaluated using various security and efficiency metrics.

This work successfully addressed the research question by exploring and testing the integration of DPF with frequency-based ranked search can indeed enhance both the efficiency and security of cloud storage services. The system could acheive improved search relevance without compromising the security significantly, even though computational overhead were observed under acceptible limits. The system offers a practical solution for organisations that require secure and efficient data retrieval, such as healthcare, finance, and legal services.

Some of the limitations are that this study does not capture the real world data complexities and network latency considerations in the experimental design. Also, simple frequency-based ranking algorithm does not cover enhanced factors that could be used for search relevance, such as context or semantic meaning.

To address these limitations, future work could be :

- Future research could include diverse, real-world datasets to assess the system's performance better in different real-world scenarios.
- Integrate more advanced ranking methods like machine learning-based algorithms could improve the accuracy and relevance of search results while also addressing the observed trade-offs in efficiency.
- Test system's scalability with larger datasets and in environments with network latency in order to obtain a more clear evaluation of its performance in practical, distributed cloud storage environments.

References

Boyle, E., Gilboa, N. & Ishai, Y. (2016), Function secret sharing: Improvements and extensions, *in* 'Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security', pp. 1292–1303.

Chamani, J. G., Wang, Y., Papadopoulos, D., Zhang, M. & Jalili, R. (2021), 'Multi-user dynamic searchable symmetric encryption with corrupted participants', *IEEE Transactions on Dependable and Secure Computing* **20**(1), 114–130.

Choi, S. G., Dachman-Soled, D., Gordon, S. D., Liu, L. & Yerukhimovich, A. (2021), Compressed oblivious encoding for homomorphically encrypted search, *in* 'Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security', pp. 2277–2291.

Dauterman, E., Feng, E., Luo, E., Popa, R. A. & Stoica, I. (2020), {DORY}: An encrypted search system with distributed trust, *in* '14th USENIX Symposium on Operating Systems Design and Implementation (OSDI 20)', pp. 1101–1119.

Gui, Z., Paterson, K. G. & Patranabis, S. (2023), Rethinking searchable symmetric encryption, *in* '2023 IEEE Symposium on Security and Privacy (SP)', IEEE, pp. 1401–1418.

Gupta, I., Singh, A. K., Lee, C.-N. & Buyya, R. (2022), 'Secure data storage and sharing techniques for data protection in cloud environments: A systematic review, analysis, and future directions', *IEEE Access* **10**, 71247–71277.

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

Khalil, M. K., Al Jahdhami, M. & Dattana, V. (2023), 'Cloud storage security compliance: An analysis of standards and regulations', *Journal of Student Research* .

Kornaropoulos, E. M., Papamanthou, C. & Tamassia, R. (2020), The state of the uniform: Attacks on encrypted databases beyond the uniform query distribution, *in* '2020 IEEE Symposium on Security and Privacy (SP)', IEEE, pp. 1223–1240.

Kumar, A., Jain, V. & Yadav, A. (2020), A new approach for security in cloud data storage for iot applications using hybrid cryptography technique, *in* '2020 International Conference on Power Electronics IoT Applications in Renewable Energy and its Control (PARC)', pp. 514–517.

Miao, Y., Deng, R. H., Liu, X., Choo, K.-K. R., Wu, H. & Li, H. (2019), 'Multi-authority attribute-based keyword search over encrypted cloud data', *IEEE Transactions on Dependable and Secure Computing* **18**(4), 1667–1680.

Stefanov, E., Dijk, M. v., Shi, E., Chan, T.-H. H., Fletcher, C., Ren, L., Yu, X. & Devadas, S. (2018), 'Path oram: an extremely simple oblivious ram protocol', *Journal of the ACM (JACM)* **65**(4), 1–26.

Sun, S.-F., Zuo, C., Liu, J. K., Sakzad, A., Steinfeld, R., Yuen, T. H., Yuan, X. & Gu, D. (2020), 'Non-interactive multi-client searchable encryption: Realization and implementation', *IEEE Transactions on Dependable and Secure Computing* **19**(1), 452–467.

Wang, Y. & Papadopoulos, D. (2021), Multi-user collusion-resistant searchable encryption with optimal search time, *in* 'Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security', pp. 252–264.

Wasabi Technologies, Inc. (2024), '90% of enterprises globally plan to increase public cloud storage budgets in 2024', https://www.businesswire.com/news/home/20240228960179/en/90-of-Enterprises-Globally-Plan-to-Increase-Public-Cloud-Storage-Budgets-in-2024. Accessed: 11 August 2024.

Yang, P., Xiong, N. & Ren, J. (2020), 'Data security and privacy protection for cloud storage: A survey', *IEEE Access* **8**, 131723–131740.

Zarezadeh, M., Mala, H. & Ashouri-Talouki, M. (2020), 'Multi-keyword ranked searchable encryption scheme with access control for cloud storage', *Peer-to-Peer Networking and Applications* **13**(1), 207–218.