Aamira Bushra, 2025, 13:6
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

International Journal of Science, Engineering and Technology
An Open Access Journal

Design and Implementation of a Decentralized Peer-to-Peer Cloud Storage System Using Kademlia Distributed Hash Table

Aamira Bushra
Department of Computer Science and Engineering Dayananda Sagar Academy of Technology and Management Bengaluru, India

Abstract - Centralized cloud storage platforms suffer from inherent limitations such as single points of failure, data privacy concerns, and dependency on trusted service providers. Decen- tralized peer-to-peer (P2P) storage systems aim to overcome these issues by distributing data across multiple independent nodes without centralized control. This paper presents the design and implementation of a decentralized cloud storage system developed in Java using the Kademlia Distributed Hash Table (DHT) for scalable peer discovery and routing. Files are divided into fixed-size chunks, encrypted locally, and distributed across participating peers using content addressing based on SHA- 256 hashes. The system supports decentralized node discovery, encrypted chunk storage, metadata-based file reconstruction, and fault tolerance through replication. Experimental evaluation on a local multi-peer environment demonstrates efficient lookup performance and improved data availability under limited peer churn.

Index Terms - Distributed Hash Table, Kademlia, Peer-to-Peer Storage, Decentralized Systems, Content Addressing, Encryption.

I. INTRODUCTION

Cloud storage services such as Google Drive and Dropbox rely on centralized server infrastructures to manage data stor- age and access. While these platforms provide scalability and ease of use, they introduce concerns related to data privacy, censorship, vendor lock-in, and single points of failure. A failure or compromise of centralized infrastructure can affect a large number of users simultaneously.

Decentralized peer-to-peer (P2P) storage systems distribute data across independent nodes, improving fault tolerance and user control. However, decentralization requires efficient mechanisms for peer discovery and data location. Distributed Hash Tables (DHTs) address these challenges by enabling scalable key-based routing.
Kademlia is a widely used DHT protocol that employs an XOR-based distance metric and structured routing tables, allowing efficient and scalable lookup operations [1]. This pa- per presents a student-level implementation of a decentralized cloud storage system using Kademlia.

II. RELATED WORK

The concept of Distributed Hash Tables has been extensively explored in peer-to-peer research. The Kademlia protocol introduced XOR-based routing to achieve logarithmic lookup complexity [1].

IPFS introduced content-addressed storage using crypto- graphic hashing to ensure data integrity and enable dedu- plication [2]. BitTorrent employs a Kademlia-based DHT for decentralized peer discovery in large-scale file-sharing systems. Blockchain platforms such as Ethereum use DHT-inspired peer discovery mechanisms to maintain decentralized communication between nodes.

Open-source Kademlia implementations available on GitHub were studied strictly as conceptual references. No external source code was reused

directly in the implementation presented in this paper.

III. SYSTEM ARCHITECTURE

The proposed system consists of two major layers: a Kademlia overlay network and a storage and application layer.

Kademlia Overlay Network
Each peer is assigned a unique node identifier and main- tains a routing table organized into k-buckets based on XOR distance ranges. The overlay supports the core Kademlia operations:

- Find Node
- Find Value
- Store

Routing tables are updated dynamically as peers communi- cate.

Storage and Application Layer
The storage layer handles file chunking, encryption, meta- data management, and interaction with the DHT. Files are split into fixed-size chunks and encrypted locally before distribu- tion. Metadata containing the ordered list of chunk hashes is retained by the file owner.

Data Storage and Security Design
Chunking and Content Addressing
Each file is divided into fixed-size chunks. A SHA-256 hash is computed for each encrypted chunk, which serves as a unique identifier within the DHT. This enables deterministic retrieval and integrity verification.

Encryption
Encryption is performed locally before chunk distribution. Since peers store only encrypted data, confidentiality is pre- served even in an untrusted storage environment.

Replication
Chunks are replicated across multiple peers to improve availability and fault tolerance when peers leave the network.

Kademlia Operations
Node Lookup
The FIND NODE operation locates peers closest to a target node identifier using iterative lookup.

Value Storage and Retrieval
The STORE operation associates chunk hashes with peers storing the corresponding data. FIND VALUE retrieves either the stored data location or the closest peers if the data is not found.

Experimental Evaluation
The system was evaluated in a local environment with multiple peer instances running on different ports. Perfor- mance metrics included lookup latency, chunk retrieval success rate, and system behavior under limited peer churn. Results demonstrate logarithmic lookup performance consistent with Kademlia theory and improved availability due to replication.

Limitations and Future Work
The current system does not include Sybil resistance or reputation mechanisms and has not been evaluated at large scale. Future work includes adaptive replication strategies, encrypted metadata sharing, NAT traversal, and large-scale deployment testing.

IV. CONCLUSION

This paper presented the design and implementation of a decentralized peer-to-peer cloud storage system using the Kademlia Distributed Hash Table. By integrating encrypted, content-addressed chunk storage with scalable DHT-based routing, the system eliminates reliance on centralized servers while maintaining data integrity and availability. The project demonstrates the feasibility of decentralized storage systems in an academic setting.

Acknowledgement
The author thanks open-source communities and academic researchers whose work provided conceptual guidance for this project.

REFERENCES

1. P. Maymounkov and D. Mazieres, "Kademlia: A Peer-to-Peer Informa- tion System Based on the XOR Metric," in Proc. Int. Workshop on Peer-to-Peer Systems (IPTPS), 2002.
2. J. Benet, "IPFS - Content Addressed, Versioned, Peer-to-Peer File System," arXiv preprint arXiv:1407.3561, 2014.
3. B. Cohen, "Incentives Build Robustness in BitTorrent," in Workshop on Economics of Peer-to-Peer Systems, 2003.
4. Ethereum Foundation, "Ethereum Peer-to-Peer Networking," 2015.
5. National Institute of Standards and Technology, "Secure Hash Standard (SHA-256)," FIPS PUB 180-4, 2015.